

BACKWARD JEU DE TAQUIN SLIDES FOR COMPOSITION TABLEAUX AND A NONCOMMUTATIVE PIERI RULE

VASU V. TEWARI

ABSTRACT. We give a backward jeu de taquin slide analogue on semistandard reverse composition tableaux. These tableaux were first studied by Haglund, Luoto, Mason and van Willigenburg when defining quasisymmetric Schur functions. Our algorithm for performing backward jeu de taquin slides on semistandard reverse composition tableaux results in a natural operator on compositions that we call the jdt operator. This operator in turn gives rise to a new poset structure on compositions whose maximal chains we enumerate. As an application, we also give a noncommutative Pieri rule for noncommutative Schur functions that uses the jdt operators.

CONTENTS

1. Introduction	2
Acknowledgements	3
2. Background on symmetric functions	3
2.1. Partitions	4
2.2. Semistandard reverse tableaux	5
2.3. Symmetric functions	5
2.4. Jeu de taquin slides on SSRTs	6
2.5. Rectification of SRTs	8
3. Background on noncommutative symmetric functions	10
3.1. Compositions	10
3.2. Semistandard reverse composition tableaux	11
3.3. Noncommutative symmetric functions	13
3.4. Noncommutative Schur functions	13
4. Box removing operators and jeu de taquin operators on compositions	14
5. Statement of main results	15
5.1. Backward jeu de taquin slides for SSRCTs	15
5.2. Right Pieri rule	18
6. Proof of validity of algorithm μ_i	19
6.1. Skew shape after μ	33

2010 *Mathematics Subject Classification.* Primary 05E05, 20C30.

Key words and phrases. Jeu de taquin, noncommutative symmetric function, Schur function, tableau, Pieri rule.

7. Backward jeu de taquin slides for SSRCTs of skew shape	34
8. A new poset on compositions	38
8.1. Growth words and SRTs	38
8.2. A poset on compositions	39
8.3. Inner shape after a sequence of slides	42
9. A right Pieri rule for noncommutative Schur functions	43
Appendix	45
References	47

1. INTRODUCTION

In this article we introduce an analogue of a combinatorial procedure, called jeu de taquin, for semistandard reverse composition tableaux and study the implications in the setting of noncommutative symmetric functions. Classically, jeu de taquin, literally the ‘teasing game’, is a set of sliding rules defined on semistandard Young tableaux (or semistandard reverse tableaux) that preserves the property of being semistandard while changing the underlying shape. This procedure is of utmost importance in algebraic combinatorics, representation theory and algebraic geometry. It was introduced by Schützenberger [18], and its many remarkable properties have been studied in depth since [3, 17, 20]. The procedure plays a crucial role in the combinatorics of permutations and Young tableaux, with deep links to the Robinson-Schensted algorithm and the plactic monoid. One of the most important applications of Schützenberger’s jeu de taquin is in giving a combinatorial rule for computing the tensor product of two representations of the symmetric group, commonly called the Littlewood-Richardson rule.

Analogues of jeu de taquin have been found for shifted tableaux [7, 16], Littelmann’s crystal paths [9], increasing tableaux [22], edge labeled Young tableaux [23] and d-complete posets [14, 15]. Procedures such as promotion and evacuation, that are consequences of jeu de taquin and important in their own right, have also been studied [13, 21]. Recently, an infinite version of jeu de taquin has been studied for infinite Young tableaux [19].

Our focus here is on semistandard reverse composition tableaux. These tableaux arise naturally in the work of Haglund, Luoto, Mason and van Willigenburg [6] for defining a distinguished basis of the Hopf algebra of quasisymmetric functions, called the basis of quasisymmetric Schur functions. These functions are indexed by compositions and are obtained from a certain specialization of nonsymmetric Macdonald polynomials [5]. The dual basis elements corresponding to them, which belong to the Hopf algebra of noncommutative symmetric functions, are noncommutative Schur functions introduced in [2]. These functions are noncommutative lifts of the classical Schur functions, and resemble them by possessing properties that are noncommutative analogues of Pieri rules, Kostka numbers, and Littlewood-Richardson rule.

A special case of the noncommutative Littlewood-Richardson rule proved in [2] is a noncommutative analogue of the classical Pieri rules. These correspond to expanding the product of noncommutative Schur functions $\mathbf{s}_{(n)} \cdot \mathbf{s}_\alpha$ and $\mathbf{s}_{(1,\dots,1)} \cdot \mathbf{s}_\alpha$ in the basis of noncommutative Schur functions, where n denotes a positive integer and α denotes a composition. Given that we are computing products in a noncommutative Hopf algebra, it is natural to consider the products $\mathbf{s}_\alpha \cdot \mathbf{s}_{(n)}$ and $\mathbf{s}_\alpha \cdot \mathbf{s}_{(1,\dots,1)}$. The statement of the noncommutative Littlewood-Richardson rule in [2], which allows us to compute the aforementioned products, requires rectification of standard reverse tableaux and a map defined by Mason in [12], denoted by ρ , that links semistandard reverse tableaux and semistandard reverse composition tableaux. One could ask whether it is possible to eliminate the use of Mason's map and introduce an analogue of rectification for semistandard reverse composition tableaux. This was achieved by Bechard in [1]. For what we aspire to compute, it is preferable to consider a procedure that is the inverse of Bechard's rectification.

The outline of this article is as follows. After reminding the reader of some classical objects from the theory of symmetric functions in Section 2, we proceed to describe the noncommutative analogues of the same in the algebra of noncommutative symmetric functions, in Section 3. In the same section, we introduce the distinguished basis of noncommutative Schur functions after introducing semistandard reverse composition tableaux. We introduce certain operators on compositions, namely box-removing operators and jdt operators, in Section 4. We state our two main results in Section 5. First, the procedure for executing the backward jeu de taquin slides on semistandard reverse composition tableaux is outlined using Algorithms 5.1 and 5.2. Second, a multiplicity-free expansion for the products $\mathbf{s}_\alpha \cdot \mathbf{s}_{(n)}$ and $\mathbf{s}_\alpha \cdot \mathbf{s}_{(1,\dots,1)}$ in terms of the noncommutative Schur functions, namely the right Pieri rule, is stated in Theorem 5.5. We then proceed to give the proofs of validity of the algorithms described earlier in Section 6 and the connection is made precise in Theorem 6.21. Next, in Subsection 6.1, we study the effect of the slides on underlying composition shapes using jdt operators. In Section 7, we describe a procedure to perform backward jeu de taquin slides on skew semistandard reverse composition tableaux. Using the jdt operators, in Section 8 we endow the set of compositions with a new poset structure \mathcal{R}_c . Finally, in Section 9, we prove the right Pieri rule in Theorem 9.3 and enumerate the maximal chains in \mathcal{R}_c in Theorem 9.4.

ACKNOWLEDGEMENTS

The author would like to thank Stephanie van Willigenburg for helpful guidance. He also wishes to thank Melissa Bechard, Matthieu Josuat-Vergès, Sarah Mason, and Ed Richmond for helpful conversations.

2. BACKGROUND ON SYMMETRIC FUNCTIONS

We will start by defining some combinatorial structures that we will encounter frequently. All the notions introduced in this section are covered in detail in [11, 17, 20].

2.1. Partitions. A *partition* λ is a finite list of positive integers $(\lambda_1, \dots, \lambda_k)$ satisfying $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k$. The integers appearing in the list are called the *parts* of the partition.

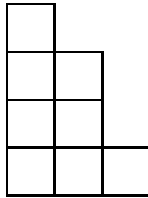
Given a partition $\lambda = (\lambda_1, \dots, \lambda_k)$, the *size* $|\lambda|$ is defined to be $\sum_{i=1}^k \lambda_i$. The number of parts of λ is called its *length*, and is denoted by $l(\lambda)$. If λ is a partition satisfying $|\lambda| = n$, then we denote this by $\lambda \vdash n$. By convention, there is a unique partition of size and length equalling 0, and we denote it by \emptyset .

We will depict a partition using its *Young diagram*. Given a partition $\lambda = (\lambda_1, \dots, \lambda_k) \vdash n$, the Young diagram of λ , also denoted by λ , is the left-justified array of n boxes, with λ_i boxes in the i -th row. We will be using the French convention where the rows are numbered from bottom to top and the columns from left to right. We refer to the box in the i -th row and j -th column by the ordered pair (i, j) . If λ and μ are partitions such that $\mu \subseteq \lambda$, that is, $l(\mu) \leq l(\lambda)$ and $\mu_i \leq \lambda_i$ for all $i = 1, 2, \dots, l(\mu)$, then the *skew shape* λ/μ is obtained by removing the first μ_i boxes from the i -th row of the Young diagram of λ for $1 \leq i \leq l(\mu)$. Given a skew shape λ/μ , we call μ the *inner shape* and λ the *outer shape*. We refer to λ as the *outer shape* and to μ as the *inner shape*. If the inner shape is \emptyset , instead of writing λ/\emptyset , we just write λ and call λ a *straight shape*. The *size* of a skew shape λ/μ , denoted by $|\lambda/\mu|$, is the number of boxes in the skew shape, which is $|\lambda| - |\mu|$.

Next, we define addable nodes and removable nodes of a partition. An *addable node* of a partition λ is a position (i, j) where a box can be appended to a part of λ so that the resulting shape is still a partition. Note that the addable nodes of a partition do not belong to the partition. On the other hand, a *removable node* of λ is a position (i, j) where a box can be removed from a part of λ so that the resulting shape is still a partition. Unlike addable nodes, removable nodes of a partition belong to the partition. Furthermore, given a partition, both addable nodes and removable nodes are uniquely determined by the column to which they belong. In the definitions just given, we have identified the partition λ with its Young diagram, also called λ .

We will define a lattice structure on the set of partitions, called *Young's lattice*, as follows. If μ is a partition obtained by adding a box at an addable node of the partition λ , we say that μ covers λ and we denote it by $\lambda \prec \mu$. The partial order obtained by taking the transitive closure of \prec allows to endow the set of partitions with the structure of a lattice, that we will denote \mathcal{Y} .

Example 2.1. The partition $\lambda = (3, 2, 2, 1)$ is represented by the following Ferrers diagram.



The box in the lower left corner is indexed by the pair $(1, 1)$. The addable nodes of λ are $(1, 4)$, $(2, 3)$, $(4, 2)$ and $(5, 1)$ while the removable nodes are $(1, 3)$, $(3, 2)$ and $(4, 1)$. Also, the partition $\mu = (3, 3, 2, 1)$ covers λ in \mathcal{Y} , that is, $(3, 2, 2, 1) \prec (3, 3, 2, 1)$.

2.2. Semistandard reverse tableaux. In this subsection, we will introduce certain classical objects that play a central role in the theory of symmetric functions.

Definition 2.2. *Given a skew shape λ/μ , a semistandard reverse tableau (SSRT) T of shape λ/μ is a filling of the boxes of λ/μ with positive integers, satisfying the condition that the entries in T are weakly decreasing along each row read from left to right and strictly decreasing along each column read from bottom to top.*

We will denote the set of all SSRTs of shape λ/μ by $\text{SSRT}(\lambda/\mu)$. Given an SSRT T , the entry in box (i, j) is denoted by $T_{(i, j)}$. The shape underlying T is denoted by $\text{sh}(T)$.

A *standard reverse tableau* (SRT) T of shape λ/μ is an SSRT that contains every positive integer in $[\lambda/\mu] = \{1, 2, \dots, |\lambda/\mu|\}$ exactly once.

The *column reading word* of an SSRT is the word obtained by reading the entries of every column in increasing order from left to right.

Let $\{x_1, x_2, \dots\}$ be an alphabet comprising of countably many commuting indeterminates. Given any SSRT T of shape $\lambda \vdash n$, we will associate a monomial x^T with it as follows.

$$x^T = \prod_{(i, j) \in \lambda} x_{T_{(i, j)}}$$

Example 2.3. *Shown below are an SSRT T of shape $(4, 3, 3, 1)$ and its associated monomial.*

$$T = \begin{array}{|c|c|c|c|} \hline 1 & & & \\ \hline 3 & 3 & 2 & \\ \hline 6 & 5 & 4 & \\ \hline 7 & 7 & 6 & 3 \\ \hline \end{array} \quad x^T = x_7^2 x_6^2 x_5 x_4 x_3^3 x_2 x_1$$

The column reading word of T is 1367 357 246 3.

2.3. Symmetric functions. The algebra of symmetric functions, denoted by Λ , is the algebra freely generated over \mathbb{Q} by countably many commuting variables $\{h_1, h_2, \dots\}$. Assigning degree i to h_i (and then extending this multiplicatively) allows us to endow Λ with the structure of a graded algebra. A basis for the degree n component of Λ , denoted by Λ^n , is given by the *complete homogeneous symmetric functions* of degree n ,

$$\{h_\lambda = h_{\lambda_1} \cdots h_{\lambda_k} : \lambda = (\lambda_1, \dots, \lambda_k) \vdash n\}.$$

A concrete realization of Λ is obtained by embedding $\Lambda = \mathbb{Q}[h_1, h_2, \dots]$ in $\mathbb{Q}[[x_1, x_2, \dots]]$, that is, the ring of formal power series in countably many commuting indeterminates $\{x_1, x_2, \dots\}$, under the identification (extended multiplicatively)

$$h_i \longmapsto \text{sum of all distinct monomials in } x_1, x_2, \dots \text{ of degree } i.$$

This viewpoint allows us to consider symmetric functions as formal power series f in the x variables with the property that $f(x_{\pi(1)}, x_{\pi(2)}, \dots) = f(x_1, x_2, \dots)$ for every permutation π of the positive integers \mathbb{N} .

The *Schur function* indexed by the partition λ , s_λ , is defined as follows.

$$s_\lambda = \sum_{T \in \text{SSRT}(\lambda)} x^T$$

While not immediate from the definition above, it is a fact that s_λ is a symmetric function. Additionally, the elements of the set $\{s_\lambda : \lambda \vdash n\}$ form a basis of Λ^n for any positive integer n .

2.4. Jeu de taquin slides on SSRTs. In this subsection, we will describe backward jeu de taquin slides in the setting of semistandard reverse tableaux. The exposition here follows that in [17] analogously.

Let $\lambda \vdash n$ and $T \in \text{SSRT}(\lambda)$. Given $i_0, j_0 \geq 1$ such that λ has an addable node in position $c = (i_0, j_0)$, a *backward jeu de taquin slide* on T starting from c gives an SSRT $\text{jdt}_{j_0}(T)$ in the manner outlined below.

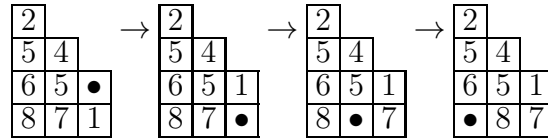
- (1) Let $c = (i_0, j_0)$.
- (2) **While** c is not equal to $(1, 1)$ **do**
 - If $c = (i, j)$, then let c' be the box of $\min(T_{(i-1, j)}, T_{(i, j-1)})$. If only one of $T_{(i-1, j)}$ and $T_{(i, j-1)}$ exists then the minimum is taken to be that single value. Furthermore, if $T_{(i-1, j)} = T_{(i, j-1)}$, then $c' = (i-1, j)$.
 - Slide $T_{c'}$ into position c . Let $c := c'$.
- (3) The final skew SSRT obtained is $\text{jdt}_{j_0}(T)$.

We give an example demonstrating the above algorithm.

Example 2.4. Let $\lambda = (3, 2, 2, 1)$ and let $T \in \text{SSRT}(\lambda)$ be

2		
5	4	
6	5	
8	7	1

If we start a backward jeu de taquin slide from the addable node given by $c = (2, 3)$, then the algorithm executes in the following manner.



Thus, we have that

$$\text{jdt}_3(T) = \begin{array}{|c|c|c|} \hline 2 & & \\ \hline 5 & 4 & \\ \hline 6 & 5 & 1 \\ \hline & 8 & 7 \\ \hline \end{array}.$$

The above algorithm can be generalized naturally to SSRT of skew shape. Now let T denote an SSRT of skew shape λ/μ . Suppose $i_0, j_0 \geq 1$ are such that the position $c = (i_0, j_0)$ is an addable node of λ . Then $\text{jdt}_{j_0}(T)$ is obtained as outlined below.

- (1) Let $c = (i_0, j_0)$.

- (2) **While** c is not an addable node of μ **do**
- If $c = (i, j)$, then let c' be the box of $\min(T_{(i-1,j)}, T_{(i,j-1)})$. If only one of $T_{(i-1,j)}$ and $T_{(i,j-1)}$ exists then the minimum is taken to be that single value. Furthermore, if $T_{(i-1,j)} = T_{(i,j-1)}$, then $c' = (i-1, j)$.
 - Slide $T_{c'}$ into position c . Let $c := c'$.
- (3) The final skew SSRT obtained is $\text{jdt}_{j_0}(T)$.

Example 2.5. Consider the SSRT T of shape $(5, 4, 3, 1)/(3, 2)$ as shown below.

1				
3	1	1		
•	•	2	1	
•	•	•	4	3

Suppose we want to compute $\text{jdt}_4(T)$. Then the following sequence of sliding moves occurs, where the box shaded red denotes the box into which entries slide.

1				
3	1	1	•	
•	•	2	1	
•	•	•	4	3

 \longrightarrow

1				
3	1	1	1	
•	•	2	•	
•	•	•	4	3

 \longrightarrow

1				
3	1	1	1	
•	•	•	2	
•	•	•	4	3

Thus, we have

$$\text{jdt}_4(T) = \begin{array}{cccc} 1 & & & \\ 3 & 1 & 1 & 1 \\ & & 2 & \\ & & 4 & 3 \end{array}.$$

For completeness, we will discuss the *forward jeu de taquin* slide next. It is the inverse of the backward jeu de taquin slide discussed above. Let T denote an SSRT of skew shape λ/μ . Suppose $i_0, j_0 \geq 1$ are such that the position $c = (i_0, j_0)$ is a removable node of μ . Then the SSRT obtained after a forward jeu de taquin slide is initiated from position c , which we denote by $\text{jdt}_{j_0}^f(T)$, is obtained as outlined below.

- (1) Let $c = (i_0, j_0)$.
- (2) **While** c is not an addable node of λ **do**
- If $c = (i, j)$, then let c' be the box of $\max(T_{(i+1,j)}, T_{(i,j+1)})$. If only one of $T_{(i+1,j)}$ and $T_{(i,j+1)}$ exists then the maximum is taken to be that single value. Furthermore, if $T_{(i+1,j)} = T_{(i,j+1)}$, then $c' = (i+1, j)$.
 - Slide $T_{c'}$ into position c . Let $c := c'$.
- (3) The final SSRT obtained is $\text{jdt}_{j_0}^f(T)$.

Example 2.6. Let T denote the SSRT below.

1				
3	1	1	1	
•	•	•	2	
•	•	•	4	3

Then $\text{jdt}_3^f(T)$ is obtained by reversing the steps in Example 2.5 and we get that

$$\text{jdt}_3^f(T) = \begin{array}{|c|c|c|c|} \hline 1 & & & \\ \hline 3 & 1 & 1 & \\ \hline \bullet & \bullet & 2 & 1 \\ \hline \bullet & \bullet & \bullet & 4 & 3 \\ \hline \end{array}.$$

2.5. Rectification of SRTs. A procedure that is closely tied to the jeu de taquin slide is rectification. To describe it in a way that is convenient for us, we need to introduce a slight variant of the Robinson- Schensted algorithm.

A discussion of the original algorithm is present in the appendix. Our variant establishes a bijection between permutations in \mathfrak{S}_n and pairs of SRTs of the same shape. We will need the notion of a *partial tableau* that we define to be an SSRT with distinct entries. Furthermore, we will abuse notation and use \emptyset to denote the empty tableau.

We will outline the algorithm next. Our input is a permutation $\sigma \in \mathfrak{S}_n$, and the output is a pair of SRTs of the same shape and size n .

Algorithm 2.7 (Variant of RS insertion algorithm).

- (1) Let $P_0 = Q_0 = \emptyset$.
- (2) **For** $i = 1$ to n , let $y := \sigma(i)$ and **do**
 - (a) Set $r := \text{first row of } P_{i-1}$.
 - (b) **While** y is greater than some number in row r , **do**
 - Let x be the greatest number smaller than y and replace x by y in P_{i-1} .
 - Set $y := x$ and $r := \text{the next row}$.
 - (c) Now that y is smaller than every number in row r , place y at the end of row r and call the resulting partial tableau P_i . If (a, b) is where y got placed, the partial tableau Q_i is obtained by placing $n - i + 1$ in position (a, b) in Q_{i-1} .
- (3) Finally, set $P := P_n$ and $Q := Q_n$.

The SRTs P and Q thus obtained are called the *insertion tableau* and *recording tableau* respectively. We present an example next.

Example 2.8. Consider the permutation $\sigma = 3157624$ in one line notation. Then the insertion algorithm works as shown below, where at each stage we note the pair (P_i, Q_i) .

$$\begin{aligned} (\emptyset, \emptyset) &\mapsto \left(\begin{array}{|c|} \hline 3 \\ \hline \end{array}, \begin{array}{|c|} \hline 7 \\ \hline \end{array} \right) \mapsto \left(\begin{array}{|c|c|} \hline 3 & 1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline 7 & 6 \\ \hline \end{array} \right) \mapsto \left(\begin{array}{|c|c|} \hline 3 & 5 \\ \hline 5 & 1 \\ \hline \end{array}, \begin{array}{|c|c|} \hline 7 & 6 \\ \hline \end{array} \right) \mapsto \left(\begin{array}{|c|} \hline 3 \\ \hline 5 \\ \hline 7 & 1 \\ \hline \end{array}, \begin{array}{|c|} \hline 4 \\ \hline 5 \\ \hline 7 & 6 \\ \hline \end{array} \right) \mapsto \left(\begin{array}{|c|} \hline 3 \\ \hline 5 & 1 \\ \hline 7 & 6 \\ \hline \end{array}, \begin{array}{|c|} \hline 4 \\ \hline 5 & 3 \\ \hline 7 & 6 \\ \hline \end{array} \right) \\ &\mapsto \left(\begin{array}{|c|c|c|} \hline 3 & & \\ \hline 5 & 1 & \\ \hline 7 & 6 & 2 \\ \hline \end{array}, \begin{array}{|c|c|c|} \hline 4 & & \\ \hline 5 & 3 & \\ \hline 7 & 6 & 2 \\ \hline \end{array} \right) \mapsto \left(\begin{array}{|c|c|} \hline 3 & 1 \\ \hline 5 & 2 \\ \hline 7 & 6 & 4 \\ \hline \end{array}, \begin{array}{|c|c|} \hline 4 & 1 \\ \hline 5 & 3 \\ \hline 7 & 6 & 2 \\ \hline \end{array} \right) \end{aligned}$$

Now, given an SRT T , let w_T denote its column reading word. Note that w_T is also a permutation written in single line notation. The *rectification* of T , denoted $\text{rect}(T)$ is defined to be $P(w_T)$, that is, the insertion tableau obtained on performing the above mentioned variant of Robinson-Schensted algorithm on w_T .

Example 2.9. Consider the SRT T of skew shape given below.

3	1		
	5		
	7	6	2
			4

Then the column reading word is 3 157 6 24. The insertion tableau for this reading word has already been computed in Example 2.8, and thus $\text{rect}(T)$ is as follows.

3	1		
5	2		
7	6	4	

An important property of rectification that will prove crucial for us is the fact that it behaves nicely with respect to jeu de taquin slides. More precisely, if T' is some SRT obtained by performing a backward jeu de taquin slide on some SRT T , then $\text{rect}(T) = \text{rect}(T')$. Rectification also allows us to describe another important combinatorial procedure called evacuation which we describe next.

Given an SRT T , the *evacuation* action on T , denoted by $e(T)$, is obtained using the algorithm below.

Algorithm 2.10 (Evacuation).

- (1) Let U be a Ferrers diagram of shape λ , where $\lambda = \text{sh}(T) \vdash n$.
- (2) Let the entry in the box in position $(1, 1)$ in T be a . Remove this entry, and compute the rectification of the remaining SRT of skew shape. Equivalently, one can remove the entry in the box in position $(1, 1)$ in T and let T' be the resulting skew SRT, and compute $\text{jdt}_1^f(T')$. This viewpoint makes it clear that the underlying shape of the rectified tableau is obtained by removing a removable node from λ . Let this node be in position (i, j) .
- (3) To the box given by (i, j) in U , assign the entry $n - a + 1$.
- (4) Repeat the previous two steps until all the boxes in U are filled. Finally, $e(T) = U$.

Example 2.11. Consider the insertion tableau from Example 2.8. We demonstrate Algorithm 2.10 by drawing the tableau T and the partially filled U at each step.

<table> <tr><td>3</td><td>1</td><td></td><td></td></tr> <tr><td>5</td><td>2</td><td></td><td></td></tr> <tr><td>7</td><td>6</td><td>4</td><td></td></tr> </table>	3	1			5	2			7	6	4		, <table> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>													\mapsto	<table> <tr><td>3</td><td>1</td><td></td><td></td></tr> <tr><td>5</td><td>2</td><td></td><td></td></tr> <tr><td>6</td><td>4</td><td></td><td>1</td></tr> </table>	3	1			5	2			6	4		1	, <table> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>													\mapsto	<table> <tr><td>1</td><td></td><td></td><td>2</td></tr> <tr><td>3</td><td>2</td><td></td><td></td></tr> <tr><td>5</td><td>4</td><td></td><td>1</td></tr> </table>	1			2	3	2			5	4		1	, <table> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>													\mapsto	<table> <tr><td>1</td><td></td><td></td><td>2</td></tr> <tr><td>3</td><td></td><td></td><td>3</td></tr> <tr><td>4</td><td>2</td><td></td><td>1</td></tr> </table>	1			2	3			3	4	2		1	, <table> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td></tr> </table>													\mapsto	<table> <tr><td>1</td><td></td><td></td><td>4</td><td>2</td></tr> <tr><td>3</td><td>2</td><td></td><td></td><td>3</td></tr> <tr><td></td><td></td><td></td><td></td><td>1</td></tr> </table>	1			4	2	3	2			3					1
3	1																																																																																																																										
5	2																																																																																																																										
7	6	4																																																																																																																									
3	1																																																																																																																										
5	2																																																																																																																										
6	4		1																																																																																																																								
1			2																																																																																																																								
3	2																																																																																																																										
5	4		1																																																																																																																								
1			2																																																																																																																								
3			3																																																																																																																								
4	2		1																																																																																																																								
1			4	2																																																																																																																							
3	2			3																																																																																																																							
				1																																																																																																																							
\mapsto	<table> <tr><td>1</td><td></td><td></td><td>4</td><td>2</td></tr> <tr><td>2</td><td></td><td></td><td>3</td><td></td></tr> <tr><td></td><td></td><td></td><td>5</td><td>1</td></tr> </table>	1			4	2	2			3					5	1	\mapsto	<table> <tr><td>1</td><td></td><td></td><td>4</td><td>2</td></tr> <tr><td></td><td></td><td></td><td>6</td><td>3</td></tr> <tr><td></td><td></td><td></td><td>5</td><td>1</td></tr> </table>	1			4	2				6	3				5	1	$\mapsto \emptyset$,	<table> <tr><td></td><td></td><td></td><td>4</td><td>2</td></tr> <tr><td></td><td></td><td></td><td>6</td><td>3</td></tr> <tr><td></td><td></td><td></td><td>7</td><td>5</td><td>1</td></tr> </table>				4	2				6	3				7	5	1																																																																								
1			4	2																																																																																																																							
2			3																																																																																																																								
			5	1																																																																																																																							
1			4	2																																																																																																																							
			6	3																																																																																																																							
			5	1																																																																																																																							
			4	2																																																																																																																							
			6	3																																																																																																																							
			7	5	1																																																																																																																						

Thus, $e(T)$ is the SRT obtained in the last step above.

Next, we will note down a property of our variant of the Robinson-Schensted algorithm. The proof is presented in the Appendix.

Lemma 2.12. *Let $w \in \mathfrak{S}_n$ and suppose further that $w \mapsto (P, Q)$ using the variant of Robinson-Schensted algorithm outlined earlier. Then*

$$w^{-1} \mapsto (e(Q), e(P)).$$

3. BACKGROUND ON NONCOMMUTATIVE SYMMETRIC FUNCTIONS

3.1. Compositions. A *composition* of n is an ordered sequence of non-negative integers $\alpha = (\alpha_1, \dots, \alpha_k)$ such that $\sum_{i=1}^k \alpha_i$ equals n . The integers α_i are called the *parts* of α . Given a composition $\alpha = (\alpha_1, \dots, \alpha_k)$, the *size* $|\alpha|$ is defined to be $\sum_{i=1}^k \alpha_i$. The number of parts of α is called its *length*, and is denoted by $l(\alpha)$. As was the case with partitions, \emptyset denotes the unique composition of size 0 and length 0.

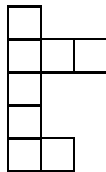
If all parts of a composition α are positive, then the composition is called a *strong* composition. If there are parts that equal 0, then we call α a *weak* composition. The notation $\alpha \models n$ denotes that α is a composition of n . We will mainly be using strong compositions (and henceforth will drop the adjective strong). If we do require weak compositions, we will state this explicitly. Given a composition α , we denote by $\tilde{\alpha}$ the partition obtained by sorting the parts of α in weakly decreasing order.

A composition $\alpha = (\alpha_1, \dots, \alpha_k)$ of n will be represented pictorially by a *reverse composition diagram* that is a left-justified array of n boxes with α_i boxes in row i . Our labeling of rows will follow the English convention. Hence the rows are labeled from top to bottom and the columns from left to right.

There exists a bijection between compositions of n and subsets of $[n-1]$ that we recall now. Given a composition $\alpha = (\alpha_1, \dots, \alpha_k) \models n$, we can associate a subset of $[n-1]$, called $\text{set}(\alpha) = \{\alpha_1, \alpha_1 + \alpha_2, \dots, \alpha_1 + \dots + \alpha_{k-1}\}$. In the opposite direction, given a set $S = \{i_1 < \dots < i_j\} \subseteq [n-1]$, we can associate a composition of n , called $\text{comp}(S) = (i_1, i_2 - i_1, \dots, i_j - i_{j-1}, n - i_j)$.

Finally, we define the *refinement order* on compositions. Given compositions α and β , we say that $\alpha \succcurlyeq \beta$ if one obtains parts of α in order by adding together adjacent parts of β in order. The composition β is said to be a *refinement* of α , or equivalently, α is said to be a *coarsening* of β .

Example 3.1. Let $\alpha = (1, 3, 1, 1, 2) \models 8$. Then $l(\alpha) = 5$ and $\text{set}(\alpha) = \{1, 4, 5, 6\} \subseteq [7]$. The reverse composition diagram of α is as shown.



Also $(4, 1, 3) \succ (1, 3, 1, 1, 2)$.

3.2. Semistandard reverse composition tableaux. Let $\alpha = (\alpha_1, \dots, \alpha_l)$ and β be compositions. Define a cover relation, \prec_c , on compositions in the following way.

$$\alpha \prec_c \beta \text{ if and only if } \begin{cases} \beta = (1, \alpha_1, \dots, \alpha_l) & \text{or} \\ \beta = (\alpha_1, \dots, \alpha_k + 1, \dots, \alpha_l) & \text{and } \alpha_i \neq \alpha_k \text{ for all } i < k \end{cases}$$

The *reverse composition poset* \mathcal{L}_c is the poset on compositions where the partial order \prec_c is obtained by taking the transitive closure of the cover relation \prec_c above. If $\beta \prec_c \alpha$, the *skew reverse composition shape* $\alpha // \beta$ is defined to be the array of boxes

$$\alpha // \beta = \{(i, j) : (i, j) \in \alpha, (i, j) \notin \beta\}$$

where β is drawn in the bottom left corner of α . We refer to α as the *outer shape* and to β as the *inner shape*. If the inner shape is \emptyset , instead of writing $\alpha // \emptyset$, we just write α and refer to α as a *straight shape*. The *size* of the skew reverse composition shape $\alpha // \beta$, denoted by $|\alpha // \beta|$, is the number of boxes in the skew reverse composition shape, that is, $|\alpha| - |\beta|$. Now, we will define a semistandard reverse composition tableau.

Definition 3.2. A semistandard reverse composition tableau (SSRCT) τ of shape $\alpha // \beta$ is a filling

$$\tau : \alpha // \beta \longrightarrow \mathbb{Z}^+$$

that satisfies the following conditions

- (1) the rows are weakly decreasing from left to right,
- (2) the entries in the first column are strictly increasing from top to bottom,
- (3) if $i < j$ and $(j, k+1) \in \alpha // \beta$ and either $(i, k) \in \beta$ or $\tau(i, k) \geq \tau(j, k+1)$ then either $(i, k+1) \in \beta$ or both $(i, k+1) \in \alpha // \beta$ and $\tau(i, k+1) > \tau(j, k+1)$.

The shape underlying an SSRCT τ will be denoted by $\text{sh}(\tau)$. We will denote the set of SSRCTs of shape $\alpha // \beta$ by $\text{SSRCT}(\alpha // \beta)$.

When considering an SSRCT, the boxes of the inner shape are filled with bullets. This given, the third condition in the definition above is equivalent to the non-existence of the following configurations in the filling τ .

$$\begin{array}{cccc} \begin{array}{c} \boxed{\bullet} \boxed{y} \\ \vdots \\ \boxed{z} \end{array} & \begin{array}{c} \boxed{\bullet} \\ \vdots \\ \boxed{z} \end{array} & \begin{array}{c} \boxed{x} \boxed{y} \\ \vdots \\ \boxed{z} \end{array} & \begin{array}{c} \boxed{x} \\ \vdots \\ \boxed{z} \end{array} \\ z \geq y > 0 & z > 0 & x \geq z \geq y > 0 & x \geq z > 0 \end{array}$$

The existence of such a configuration in a filling will be termed a *triple rule violation*. Our strategy to show that a given filling is an SSRCT will be to show that there are no triple rule violations therein.

The *column reading word* of an SSRCT τ is the word obtained by reading the entries in each column in increasing order, where we traverse the columns from left to right. We denote it by w_τ .

Example 3.3. An SSRCT of shape $(2, 5, 4, 2) \parallel (2, 1, 2)$ (left) and an SRCT of shape $(3, 4, 2, 3)$.

The left grid represents the initial state of the 15-puzzle. It is a 4x4 grid with the following values:

4	3		
•	•	7	5
•	7	6	2
•	•		

The right grid represents the state after one move. It is a 4x4 grid with the following values:

6	4	1	
8	7	5	2
10	3		
12	11	9	

Given a composition α , there exists a unique SRCT τ_α so that the underlying shape of τ_α is α and $\text{comp}(\tau_\alpha) = \alpha$. This tableau is called the *canonical tableau* of shape α . To construct τ_α given $\alpha = (\alpha_1, \dots, \alpha_k)$, place consecutive integers from $1 + \sum_{i=1}^{r-1} \alpha_i$ to $\sum_{i=1}^r \alpha_i$ in that order from right to left in the r -th row from the top in the reverse composition diagram of α for $1 \leq r \leq k$.

3	2	1	
7	6	5	4
9	8		
12	11	10	

$$\rho_\alpha : \text{SSRCT}(-//\alpha) \longrightarrow \text{SSRT}(-/\tilde{\alpha})$$

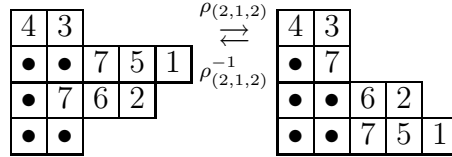
We give the description of the inverse map ρ_α^{-1} next. Let T be an SSRT.

- (1) Take the first column of the SSRT, and place the entries of this column above the first column of the inner shape α in increasing order from top to bottom.

- (2) Now take the set of entries in the second column in decreasing order and place them in the row with smallest index so that either
- the box to the immediate left of the number being placed is filled and the entry therein is weakly greater than the number being placed
 - the box to the immediate left of the number being placed belongs to the inner shape α .

As a matter of convention, instead of writing ρ_\emptyset or ρ_\emptyset^{-1} , we will use ρ or ρ^{-1} respectively.

Example 3.5. An SSRCT of shape $(2, 5, 4, 2) \parallel (2, 1, 2)$ (left), and the corresponding SSRT of shape $(5, 4, 2, 2) / (2, 2, 1)$ (right).



3.3. Noncommutative symmetric functions. An algebra closely related to Λ is the algebra of *noncommutative symmetric functions* **NSym**, introduced in [4]. It is the free associative algebra $\mathbb{Q}\langle \mathbf{h}_1, \mathbf{h}_2, \dots \rangle$ generated by a countably infinite number of indeterminates \mathbf{h}_k for $k \geq 1$. Assigning degree k to \mathbf{h}_k , and extending this multiplicatively allows us to endow **NSym** with the structure of a graded algebra. A natural basis for the degree n graded component of **NSym**, denoted by \mathbf{NSym}^n , is given by the *noncommutative complete homogeneous symmetric functions*, $\{\mathbf{h}_\alpha = \mathbf{h}_{\alpha_1} \cdots \mathbf{h}_{\alpha_k} : \alpha = (\alpha_1, \dots, \alpha_k) \models n\}$. The link between Λ and **NSym** is made manifest through the *forgetful* map, $\chi : \mathbf{NSym} \rightarrow \Lambda$, defined by mapping \mathbf{h}_i to h_i and extending multiplicatively. Thus, the images of elements of **NSym** under χ are elements of Λ .

NSym has another important basis called the noncommutative ribbon Schur basis. We will denote the *noncommutative ribbon Schur function* indexed by a composition β by \mathbf{r}_β . The following [4, Proposition 4.13] can be taken as the definition of the noncommutative ribbon Schur functions.

$$\mathbf{r}_\beta = \sum_{\alpha \succ \beta} (-1)^{l(\beta) - l(\alpha)} \mathbf{h}_\alpha$$

Using the above basis, we can define the noncommutative Schur functions.

3.4. Noncommutative Schur functions. We will now describe a distinguished basis for **NSym**, introduced in [2], called the basis of *noncommutative Schur functions*. These functions are naturally indexed by compositions, and the noncommutative Schur function indexed by a composition α will be denoted by \mathbf{s}_α . They are defined implicitly using the relation

$$\mathbf{r}_\beta = \sum_{\alpha \models |\beta|} d_{\alpha\beta} \mathbf{s}_\alpha$$

where $d_{\alpha\beta}$ is the number of SRCTs of shape α and descent composition β .

The noncommutative Schur function \mathbf{s}_α satisfies the following important property [2, Equation 2.12].

$$\chi(\mathbf{s}_\alpha) = s_{\tilde{\alpha}}$$

Thus, the noncommutative Schur functions are noncommutative lifts of the Schur functions in \mathbf{NSym} and, in fact, share many properties with the Schur functions. The interested reader should refer to [2, 10] for a detailed study of these functions. For our purposes, we require the noncommutative Littlewood-Richardson rule proved in [2].

Theorem 3.6. [2, Theorem 3.5] *Given compositions α and β , the product $\mathbf{s}_\alpha \cdot \mathbf{s}_\beta$ can be expanded in the basis of noncommutative Schur functions as follows*

$$\mathbf{s}_\alpha \cdot \mathbf{s}_\beta = \sum_{\gamma} C_{\alpha\beta}^{\gamma} \mathbf{s}_{\gamma}$$

where $C_{\alpha\beta}^{\gamma}$ counts the number of SRCTs of shape $\gamma // \beta$ that rectify to the canonical tableau of shape α .

We should clarify what it means to rectify for SRCTs. We say that an SRCT τ_1 of shape $\gamma // \beta$ rectifies to a tableau τ_2 of straight shape α if $\rho^{-1}(P(w_{\tau_1})) = \tau_2$.

4. BOX REMOVING OPERATORS AND JEU DE TAQUIN OPERATORS ON COMPOSITIONS

In this section, we will define some operators on compositions that will play an important role later.

The *box removing operators* on compositions, denoted by \mathfrak{d}_i for $i \geq 1$, have the following description: $\mathfrak{d}_i(\alpha) = \alpha'$ where α' is the composition obtained by subtracting 1 from the rightmost part of length i in α and omitting any 0s that might result in so doing. If there is no such part, then $\mathfrak{d}_i(\alpha) = 0$.

Example 4.1. Let $\alpha = (2, 1, 2)$. Then $\mathfrak{d}_1(\alpha) = (2, 2)$ and $\mathfrak{d}_2(\alpha) = (2, 1, 1)$.

Given a nonempty set of positive integers $S = \{a_1 < a_2 < \dots < a_r\}$, we will also be needing the following product of box removing operators.

$$\mathbf{v}_S = \mathfrak{d}_{a_1} \mathfrak{d}_{a_2} \dots \mathfrak{d}_{a_r}$$

Note that the product above is not commutative, and we will consider \mathbf{v}_S as operators $\mathfrak{d}_{a_r}, \mathfrak{d}_{a_{r-1}}, \dots, \mathfrak{d}_{a_1}$ applied in succession to a composition, in that order. As a matter of convention, \mathbf{v}_{\emptyset} is to be interpreted as the identity map. Of particular interest to us is the operator $\mathbf{v}_{[i]}$ for $i \geq 1$, where $[i] = \{1, \dots, i\}$. We define $[0]$ to be the empty set \emptyset . Then $\mathbf{v}_{[0]}$ is the identity map.

Next, define an operator a_i that appends a part of length i to a composition α at the end, for $i \geq 1$. For example, $a_2((2, 1, 3)) = (2, 1, 3, 2)$. Notice also that $a_j \mathfrak{d}_2((3, 5, 1)) = 0$ for any positive integer j , as $\mathfrak{d}_2((3, 5, 1)) = 0$.

With the definitions of a_i and $\mathfrak{v}_{[i]}$, we can define the *jeu de taquin operators* (henceforth abbreviated to *jdt operators*), u_i for $i \geq 1$, as follows.

$$u_i = a_i \mathfrak{v}_{[i-1]}$$

Example 4.2. We will compute $u_4(\alpha)$ where $\alpha = (6, 3, 2, 3, 1, 5, 4)$. Firstly, $\mathfrak{d}_1 \mathfrak{d}_2 \mathfrak{d}_3(\alpha) = (6, 3, 2, 1, 5, 4)$, and thus $a_4 \mathfrak{v}_{[3]}(\alpha) = (6, 3, 2, 1, 5, 4, 4)$.

Remark 4.3. Let α be a composition and $\lambda = \tilde{\alpha}$. Let $\beta = u_i(\alpha)$ for some positive integer i . Then β is the partition obtained by adding a box at the addable node to λ in column i .

5. STATEMENT OF MAIN RESULTS

In this section, we will state our algorithm for our analogue of the classical backward jeu de taquin slide. We will work under the following assumptions in this section and for all the proofs in the coming sections, unless otherwise stated.

- i will denote a positive integer ≥ 1 .
- T will refer to a fixed SSRT of shape λ .
- The SSRCT $\rho^{-1}(T)$ will be denoted by τ and we will assume that it has shape α where α clearly satisfies $\tilde{\alpha} = \lambda$.
- There exists an addable node in column $i + 1$ of λ .

5.1. Backward jeu de taquin slides for SSRCTs. With the notation from above, define a positive integer r_1 to be such that α_{r_1} is the bottommost part of length i in α . This given define positive integers r_j for $j \geq 2$ recursively subject to the following conditions.

- I. $r_j < r_{j-1}$.
- II. r_j is the greatest positive integer such that $\tau(r_j, i) > \tau(r_{j-1}, i) \geq \tau(r_j, i + 1)$. If the box in position $(r_j, i + 1)$ does not belong to τ , we let $\tau(r_j, i + 1)$ equal 0.

Clearly the r_j s defined above are finitely many. Let r_k denote the minimum element and $S = \{r_1 > \dots > r_k\}$. Next we give an algorithm ϕ_{i+1} that takes τ as input and outputs an SSRCT $\phi_{i+1}(\tau)$ using the set S above.

Algorithm 5.1.

- (i) For $j=k$ down to 1 in that order,
 - Replace the entry in box (r_j, i) by the entry in box (r_{j-1}, i) . The entry $\tau(r_k, i)$ exits the tableau in every iteration.
 - Remove the box in position (r_1, i) .
- (ii) Denote the resulting filling by $\phi_{i+1}(\tau)$.

We will use the above algorithm and give a procedure on SSRCTs of straight shape, referred to as μ_i , and this is analogous to the procedure jdt_i on SSRTs of straight shape. Our input is again τ and the output is an SSRCT $\mu_i(\tau)$.

Algorithm 5.2.

- (i) If $i \geq 2$, compute first the SSRCT $\phi_2 \circ \cdots \circ \phi_i(\tau)$. Denote this by τ' . If $i = 1$, define τ' to be τ . Furthermore, store the entries that exit in an array H , in the case $i \geq 2$.
- (ii) Let $\text{sh}(\tau') = \gamma = (\gamma_1, \dots, \gamma_k)$. Let $\delta = (\gamma_1, \dots, \gamma_k, i)$.
- (iii) Consider a filling of the skew reverse composition shape $\delta // (1)$ where the top k rows are filled according to τ' , and the last row is filled by the entries in H in decreasing order from left to right.
- (iv) The filling of the skew reverse composition shape $\delta // (1)$ obtained above is defined to be $\mu_i(\tau)$.

Now we will state a theorem that gives credence to our claim that μ_i is analogous to jdt_i .

Theorem 5.3. *With the notation as before, the following diagram commutes.*

$$\begin{array}{ccc}
 \text{SSRTs} & \xrightarrow{\rho^{-1}} & \text{SSRCTs} \\
 \text{jdt}_i \downarrow & & \downarrow \mu_i \\
 \text{SSRTs} & \xrightarrow[\rho_{(1)}^{-1}]{} & \text{SSRCTs}
 \end{array}$$

Example 5.4. *Given below is an SSRT T (left) and the SSRCT $\tau = \rho^{-1}(T)$ (right).*

$T =$	<table border="1"> <tr><td>8</td><td></td><td></td><td></td></tr> <tr><td>19</td><td>9</td><td>6</td><td></td></tr> <tr><td>24</td><td>17</td><td>7</td><td>3</td></tr> <tr><td>26</td><td>23</td><td>18</td><td>10</td></tr> <tr><td>37</td><td>28</td><td>25</td><td>20</td></tr> <tr><td>43</td><td>35</td><td>27</td><td>22</td></tr> <tr><td>44</td><td>38</td><td>36</td><td>29</td></tr> <tr><td>47</td><td>42</td><td>41</td><td>33</td></tr> <tr><td>48</td><td>46</td><td>45</td><td>40</td></tr> </table>	8				19	9	6		24	17	7	3	26	23	18	10	37	28	25	20	43	35	27	22	44	38	36	29	47	42	41	33	48	46	45	40	$\tau =$	<table border="1"> <tr><td>8</td><td></td><td></td><td></td></tr> <tr><td>19</td><td>17</td><td>7</td><td>3</td></tr> <tr><td>24</td><td>23</td><td>18</td><td>10</td></tr> <tr><td>26</td><td>9</td><td>6</td><td></td></tr> <tr><td>37</td><td>35</td><td>27</td><td>22</td></tr> <tr><td>43</td><td>42</td><td>41</td><td>40</td></tr> <tr><td>44</td><td>38</td><td>36</td><td>33</td></tr> <tr><td>47</td><td>46</td><td>45</td><td>29</td></tr> <tr><td>48</td><td>28</td><td>25</td><td>20</td></tr> </table>	8				19	17	7	3	24	23	18	10	26	9	6		37	35	27	22	43	42	41	40	44	38	36	33	47	46	45	29	48	28	25	20
8																																																																											
19	9	6																																																																									
24	17	7	3																																																																								
26	23	18	10																																																																								
37	28	25	20																																																																								
43	35	27	22																																																																								
44	38	36	29																																																																								
47	42	41	33																																																																								
48	46	45	40																																																																								
8																																																																											
19	17	7	3																																																																								
24	23	18	10																																																																								
26	9	6																																																																									
37	35	27	22																																																																								
43	42	41	40																																																																								
44	38	36	33																																																																								
47	46	45	29																																																																								
48	28	25	20																																																																								

We will compute $\mu_9(\tau)$. We start by computing $\phi_2 \circ \cdots \circ \phi_9(\tau)$. To comprehensively illustrate the theorem, this is shown step by step below. The entries in rows r_j for $1 \leq j \leq k$ that we need in applying Algorithm 5.1 are depicted in green, except the entry that exits,

which is depicted in red.

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	13	5		
43	42	41	40	39	34	30	14	11	
44	38	36	33	32	31	4	2		
47	46	45	29	16	12	1			
48	28	25	20						

$$\phi_9(\tau)$$

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	13	2		
43	42	41	40	39	34	30	14	11	
44	38	36	33	32	31	4			
47	46	45	29	16	12	1			
48	28	25	20						

$$\phi_8 \circ \phi_9(\tau)$$

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	4	2		
43	42	41	40	39	34	30	14	11	
44	38	36	33	32	31	1			
47	46	45	29	16	12				
48	28	25	20						

$$\phi_7 \circ \phi_8 \circ \phi_9(\tau)$$

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	4	2		
43	42	41	40	39	31	30	14	11	
44	38	36	33	32	12	1			
47	46	45	29	16					
48	28	25	20						

$$\phi_6 \circ \dots \circ \phi_9(\tau)$$

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	4	2		
43	42	41	40	32	31	30	14	11	
44	38	36	33	16	12	1			
47	46	45	29						
48	28	25	20						

$$\phi_5 \circ \dots \circ \phi_9(\tau)$$

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	4	2		
43	42	41	33	32	31	30	14	11	
44	38	36	29	16	12	1			
47	46	45	20						
48	28	25							

$$\phi_4 \circ \dots \circ \phi_9(\tau)$$

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	4	2		
43	42	41	33	32	31	30	14	11	
44	38	36	29	16	12	1			
47	46	25	20						
48	28								

$$\phi_3 \circ \dots \circ \phi_9(\tau)$$

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	4	2		
43	42	41	33	32	31	30	14	11	
44	38	36	29	16	12	1			
47	28	25	20						
48									

$$\phi_2 \circ \dots \circ \phi_9(\tau)$$

Figure 1 displays two triangular arrays of numbers. The left array has 8 rows, and the right array has 8 rows. The bottom row of each array is highlighted in red.

Left Array (Row 8 highlighted):

8									
19	17	7	3						
24	23	18	10						
26	9	6							
37	35	27	22	21	15	4	2		
43	42	41	33	32	31	30	14	11	
44	38	36	29	16	12	1			
47	28	25	20						
48	46	45	40	39	34	13	5		

Right Array (Row 8 highlighted):

8									
19	9	6							
24	17	7	3						
26	23	18	10						
37	28	25	20						
43	35	27	22	16	12	1			
44	38	36	29	21	15	4	2		
47	42	41	33	32	31	30	13	5	
48	46	45	40	39	34	14	11		

$$\mathbf{S}_{(1,4,2)} \cdot \mathbf{S}_{(3)} = \mathbf{S}_{(1,4,2,3)} + \mathbf{S}_{(1,2,2,5)} + \mathbf{S}_{(1,4,1,4)} + \mathbf{S}_{(1,3,1,5)} + \mathbf{S}_{(1,2,1,6)} + \mathbf{S}_{(4,2,4)} \\ + \mathbf{S}_{(3,2,5)} + \mathbf{S}_{(2,2,6)} + \mathbf{S}_{(1,4,5)} + \mathbf{S}_{(1,3,6)} + \mathbf{S}_{(1,2,7)}$$

Next, we compute $\mathbf{s}_\alpha \cdot \mathbf{s}_{(1,1,1)}$. Then we are interested in finding positive integers $i_1 \geq i_2 \geq i_3$ such that $u_{i_3}u_{i_2}u_{i_1}(\alpha)$ is a valid composition of size 10. Thus, we have the following choices.

$$\begin{aligned} u_1u_1u_1(\alpha) &= (1, 4, 2, 1, 1, 1) & u_1u_1u_2(\alpha) &= (4, 2, 2, 1, 1) & u_1u_1u_3(\alpha) &= (1, 4, 3, 1, 1) \\ u_1u_1u_5(\alpha) &= (1, 2, 5, 1, 1) & u_1u_2u_3(\alpha) &= (4, 3, 2, 1) & u_1u_2u_5(\alpha) &= (2, 5, 2, 1) \\ u_1u_3u_5(\alpha) &= (1, 5, 3, 1) & u_2u_3u_5(\alpha) &= (5, 3, 2) \end{aligned}$$

This gives us the following expansion for $\mathbf{s}_{(1,4,2)} \cdot \mathbf{s}_{(1,1,1)}$.

$$\begin{aligned} \mathbf{s}_{(1,4,2)} \cdot \mathbf{s}_{(1,1,1)} &= \mathbf{s}_{(1,4,2,1,1,1)} + \mathbf{s}_{(4,2,2,1,1)} + \mathbf{s}_{(1,4,3,1,1)} + \mathbf{s}_{(1,2,5,1,1)} + \mathbf{s}_{(4,3,2,1)} \\ &\quad + \mathbf{s}_{(2,5,2,1)} + \mathbf{s}_{(1,5,3,1)} + \mathbf{s}_{(5,3,2)} \end{aligned}$$

6. PROOF OF VALIDITY OF ALGORITHM μ_i

In this section, we will give proofs of validity for the results in Subsection 5.1. But prior to that, we will require certain results on SSRTs. As stated earlier, we will be using the notation established at the start of Section 5 for the proofs in this section.

A particular entry of T that plays a significant role in what follows is the first entry that moves horizontally while computing $\text{jdt}_{i+1}(T)$. This entry is $T_{j,i}$ where j is the largest integer such that $T_{j,i} < T_{j-1,i+1}$. (We assume $T_{0,s} = \infty$ for all positive integers s .) We shall denote this entry by $f_{i+1}(T)$ (the reason behind the name being that it is the first entry that moves horizontally). We will use this in the next couple of lemmas, whose proofs are straightforward.

Lemma 6.1. *Let $f_{i+1}(T) = T_{r,i}$. Let T' be the tableau defined below.*

$$\begin{aligned} T'_{p,q} &= T_{p,q} \text{ if } q \neq i \\ T'_{p,i} &= T_{p,i} \text{ if } p < r \\ T'_{p,i} &= T_{p+1,i} \text{ if } p \geq r. \end{aligned}$$

Then T' is an SSRT.

Remark 6.2. *In words, T' is obtained from T by first removing $T_{r,i}$ and sliding all entries above it in the i -th column down by one row. Note that the rest of the tableau is left unchanged.*

Proof. The only thing we need to check is that the entries in each row of T' are weakly decreasing when read from left to right. More specifically, it follows from the preceding remark that we only need to check if $T'_{j,i} \geq T'_{j,i+1}$ and $T'_{j,i-1} \geq T'_{j,i}$ (assuming $i \geq 2$ for this case) for $j \geq r$.

The former inequality is clear since $T'_{j,i} = T_{j+1,i}$ and $T'_{j,i+1} = T_{j,i+1}$ for $j \geq r$ and by our hypothesis $T_{r,i}$ is the first entry that moves horizontally when a backward jeu de taquin slide is initiated from the $i+1$ -th column.

Now we will show that $T'_{j,i-1} \geq T'_{j,i}$ for $j \geq r$. We have $T'_{j,i} = T_{j+1,i}$ and $T'_{j,i-1} = T_{j,i-1}$. But it is clear that $T_{j,i-1} > T_{j+1,i}$ in any SSRT T . Hence the claim follows. \square

We will denote the SSRT T' above by $\text{tjdt}_{i+1}(T)$ (the reason behind the name tjdt is that it stands for a ‘truncated’ jeu de taquin slide). Furthermore, we define $\text{tjdt}_1(T) = T$, as no entries move horizontally when a backward jeu de taquin slide is initiated from the first column.

Example 6.3. *Let*

$$T = \begin{array}{|c|c|c|} \hline 6 & & \\ \hline 7 & 6 & \\ \hline 8 & 7 & 4 \\ \hline 10 & 8 & 5 \\ \hline 11 & 9 & 9 & 8 \\ \hline \end{array}.$$

If we consider a backward jeu de taquin slide from the addable node in column 3, then we have $f_3(T) = T_{2,2} = 8$. Thus

$$\text{tjdt}_3(T) = \begin{array}{|c|c|c|} \hline 6 & & \\ \hline 7 & & \\ \hline 8 & 6 & 4 \\ \hline 10 & 7 & 5 \\ \hline 11 & 9 & 9 & 8 \\ \hline \end{array}.$$

In the computation of $\text{jdt}_{i+1}(T)$, clearly i entries in T move horizontally. It is crucial for us to know these entries. Define

$$H_{i+1}(T) = \text{multiset of entries in } T \text{ that move horizontally.}$$

We define $H_1(T)$ to be the empty set. With this definition at hand, we state a second lemma.

Lemma 6.4.

$$H_i(\text{tjdt}_{i+1}(T)) \cup \{f_{i+1}(T)\} = H_{i+1}(T).$$

In the equality above, we are considering union of multisets.

Proof. Let $T' = \text{tjdt}_{i+1}(T)$ and $f_{i+1}(T) = T_{r,i}$. Our construction of T' implies that there is an addable node in column i in the partition shape underlying T' . Hence, we can initiate a backward jeu de taquin slide from the i -th column of T' and $H_i(T')$ is well-defined.

If $i = 1$, then we have nothing to prove. Hence, we can assume that $i \geq 2$. We will show that no entry of the form $T'_{j,i-1}$ with $j > r$ moves horizontally when a backward jeu de taquin slide is initiated from column i in T' , as this clearly suffices to establish the claim.

Suppose this was not the case. Then there exists $j > r$ such that

$$T'_{j,i-1} < T'_{j-1,i}.$$

Since $T'_{j,i-1} = T_{j,i-1}$ and $T'_{j-1,i} = T_{j,i}$ for $j > r$, the inequality above implies

$$T_{j,i-1} < T_{j,i}.$$

But this contradicts the fact that T is an SSRT. □

Example 6.5. *Let*

$$T = \begin{array}{|c|c|c|c|} \hline 6 & & & \\ \hline 7 & 6 & & \\ \hline 8 & 7 & 2 & \\ \hline 10 & 8 & 5 & 3 \\ \hline 11 & 9 & 9 & 8 \\ \hline \end{array}.$$

If we consider a backward jeu de taquin slide from the addable node in column 4, then the entries in the boxes coloured green move horizontally. Thus, we have

$$H_4(T) = \{2, 8, 11\} \text{ and } f_4(T) = 2.$$

We also have that

$$\text{tjdt}_4(T) = \begin{array}{|c|c|c|c|} \hline 6 & & & \\ \hline 7 & 6 & & \\ \hline 8 & 7 & & \\ \hline 10 & 8 & 5 & 3 \\ \hline 11 & 9 & 9 & 8 \\ \hline \end{array}.$$

Now if one does a backward jeu de taquin slide on the tableau above from the third column, then the entries coloured green move horizontally, therefore $H_3(\text{tjdt}_4(T)) = \{8, 11\}$. Thus, we can verify in this case that

$$H_4(T) = H_3(\text{tjdt}_4(T)) \cup f_4(T).$$

The lemma above allows us to recursively compute all the entries that move horizontally when a backward jeu de taquin slide is performed from an addable node in column $i + 1$. We already know how to obtain $\text{tjdt}_{i+1}(T)$ given an SSRT T . Before we define a backward jeu de taquin slide for SSRCTs, let us consider the relatively easier question of computing $\rho^{-1}(\text{tjdt}_{i+1}(T))$ given τ . This is precisely what the algorithm ϕ_{i+1} described in Subsection 5.1 achieves. We will establish that ϕ_{i+1} maps SSRCTs to SSRCTs such that the following diagram commutes.

$$\begin{array}{ccc} \text{SSRTs} & \xrightarrow{\rho^{-1}} & \text{SSRCTs} \\ \text{tjdt}_{i+1} \downarrow & & \downarrow \phi_{i+1} \\ \text{SSRTs} & \xrightarrow{\rho^{-1}} & \text{SSRCTs} \end{array}$$

To allow for cleaner proofs, we will now describe how we will think of T and τ diagrammatically. We will think of T as aligned to the left in a rectangular Ferrers diagram where

$$\begin{aligned} \text{number of rows, } n &= l(\lambda), \\ \text{number of columns} &\geq \lambda_1 + 1. \end{aligned}$$

The empty boxes of this Ferrers diagram are assumed to be filled with 0s. We will call this diagram T_\square . Note that the 0s do not play any role as far as doing the backward jeu de taquin slides are concerned.

Similarly, we will think of τ as being placed left aligned in a rectangular Ferrers diagram where

$$\begin{aligned} \text{number of rows, } n &= l(\alpha), \\ \text{number of columns} &\geq \lambda_1 + 1. \end{aligned}$$

The empty boxes of this Ferrers diagram are assumed to be filled with 0s. We will assume that the dimensions of the rectangular Ferrers diagram are the same for both T and τ . The exact dimensions will not matter as long as they satisfy the constraints mentioned above. We will call the rectangular filling τ_\square .

For computing $\text{tjdt}_{i+1}(T)$, we only need to know what the entries in columns i and $i+1$ are. Since we will work with T_\square we will take the 0s into account when considering the i -th and $i+1$ -th columns. Let the i -th and $i+1$ -th columns of T_\square be as shown below.

$$\begin{array}{|c|c|} \hline a_1 & b_1 \\ \hline a_2 & b_2 \\ \hline \vdots & \vdots \\ \hline a_n & b_n \\ \hline \end{array}$$

Furthermore, let $a_{n+1} = b_{n+1} = \infty$. Since we have assumed that there is an addable node in the $i+1$ -th column of λ , we know that

$$|\{b_r : b_r = 0, 1 \leq r \leq n\}| > |\{a_r : a_r = 0, 1 \leq r \leq n\}|. \quad (1)$$

Let

$$\text{min} = \text{smallest positive integer such that } a_{\text{min}} < b_{\text{min}+1}.$$

Then, we have that

$$a_{\text{min}} = f_{i+1}(T).$$

Clearly, $a_{\text{min}} > 0$. Another trivial observation, that we will use without mentioning is that $a_p = a_q$ for some $1 \leq p, q \leq n$ if and only if $a_p = a_q = 0$.

Corresponding to T_\square above, the i -th and $i+1$ -th columns of τ_\square are as shown below.

$$\begin{array}{|c|c|} \hline c_1 & d_1 \\ \hline c_2 & d_2 \\ \hline \vdots & \vdots \\ \hline c_n & d_n \\ \hline \end{array}$$

Clearly the sequence c_1, c_2, \dots, c_n is a rearrangement of the sequence a_1, a_2, \dots, a_n . A similar statement holds for the sequences d_1, d_2, \dots, d_n and b_1, b_2, \dots, b_n .

Now let us consider what we require from the procedure ϕ_{i+1} . We would like it to locate $f_{i+1}(T)$ in τ and then remove it from τ and rearrange the remaining entries so that what we have is still an SSRCT. The algorithm to accomplish this will be presented after some

intermediate lemmas. For all the lemmas that follow, we will assume that s_1 is the positive integer such that

$$c_{s_1} = a_{min}.$$

Our first lemma implies that a_{min} does not equal any other entry in the i -th and $i + 1$ -th columns of T_\square . We know that $a_{min} > 0$. Hence it clearly can not equal any other entry in the i -th column since T is an SSRT. To see that a_{min} does not equal any entry in the $i + 1$ -th column, the following suffices.

Lemma 6.6. $a_{min} \neq b_{min}$.

Proof. Assume, contrary to what is to be established, that $a_{min} = b_{min}$. Since $a_{min} > 0$, we get that $min > 1$, as b_1 is definitely 0 given the inequality in (1). Now, since $a_{min} > a_{min-1}$, our assumption gives $a_{min-1} < b_{min}$. But this contradicts the definition of min . \square

Our next lemma deals with entries greater than $c_{s_1} = a_{min}$ in τ_\square .

Lemma 6.7. $c_t > c_{s_1} \iff d_t \geq b_{min+1}$.

Proof. Since $c_{s_1} = a_{min}$, we know that $b_{min+1} > c_{s_1}$. Thus, $b_{min+1} > c_j$ for all $1 \leq j \leq n$ satisfying $c_j \leq c_{s_1}$. Thus any entry in the $i + 1$ -th column of T_\square that is $\geq b_{min+1}$ can not be the neighbour of any entry that is $\leq c_{s_1}$ in the i -th column of τ_\square . Consider the following sets.

$$\begin{aligned} X &= \{c_j : c_j > c_{s_1}\} \\ Y &= \{b_j : b_j \geq b_{min+1}\} \end{aligned}$$

The fact that X and Y are sets, and not multisets, follows from the fact that $b_{min+1} > c_{s_1} = a_{min} > 0$. Now, we know from our argument at the beginning of this proof that an element of Y can only neighbour an element of X in τ_\square . But $|X| = |Y|$, as X is the same as $\{a_j : a_j > a_{min}\}$. Hence each element of Y is the neighbour to a unique element of X , and each element of X has a neighbour that is an element of Y . Hence the claim follows. \square

We note down an important consequence of the above lemma in the following remark.

Remark 6.8. Consider the i -th and $i + 1$ -th columns of T_\square again.

a_1	b_1
\vdots	\vdots
a_{min}	b_{min}
a_{min+1}	b_{min+1}
\vdots	\vdots
a_n	b_n

The boxes in the i -th column that contain entries $\leq a_{min}$ and the boxes in the $i + 1$ -th column that contain entries $\leq b_{min}$ are coloured green. All other boxes, that is, the boxes in the i -th column that contain entries $> a_{min}$ and the boxes in the $i + 1$ -th column that contain entries $> b_{min}$ are coloured orange. Notice that the sets X and Y defined in the proof of Lemma 6.7 correspond to the orange entries in the i -th column and $i + 1$ -th column respectively. Then, Lemma 6.7 implies that in τ_\square , the values belonging to the green boxes in T_\square get neighbours from the green boxes only, while the values that belong to the orange boxes in T_\square get neighbours from the orange boxes only.

Recall now that s_1 is the positive integer such that $c_{s_1} = a_{min}$. Define recursively integers s_j satisfying $1 \leq s_j \leq n$ for $j \geq 2$ using the following criteria:

- I. $s_j > s_{j-1}$,
- II. c_{s_j} is the greatest positive integer that satisfies

$$c_{s_{j-1}} > c_{s_j} \geq d_{s_{j-1}}.$$

Clearly, only finitely many s_j exist. Let the maximum be given by s_k and let

$$S = \{s_1, \dots, s_k\}.$$

With this setup, we have the following lemma.

Lemma 6.9. If $t > s_j$ for some $1 \leq j \leq k$ such that $c_t > c_{s_j}$, then $c_t > c_{s_1}$.

Proof. If $j = 1$, the claim is clearly true. Hence assume $j \geq 2$. Shown below is the configuration in τ_\square that we will focus on.

$$\begin{array}{cc}
 \boxed{c_{s_{j-1}}} & \boxed{d_{s_{j-1}}} \\
 \vdots & \vdots \\
 \boxed{c_{s_j}} & \boxed{d_{s_j}} \\
 \vdots & \\
 \boxed{c_t}
 \end{array}$$

Firstly, note that since $c_{s_1} > c_{s_2} > \cdots > c_{s_k}$, we can not have t being equal to any s_i for some $1 \leq i \leq k$. For if $t = s_i$ for some i , then we know that $c_t < c_{s_j}$ for all $s_j < t$. This contradicts our assumption on t .

We will now show that the hypothesis implies $c_t > c_{s_{j-1}}$, as this suffices to establish the claim. We know that

$$c_{s_{j-1}} > c_{s_j} \geq d_{s_{j-1}}. \quad (2)$$

We also know, by the definition of s_j , that $s_j > s_{j-1}$ is such that c_{s_j} is the greatest positive integer satisfying the inequalities in (2). Since $t > s_j$, we know that c_t does not satisfy

$$c_{s_{j-1}} > c_t \geq d_{s_{j-1}}. \quad (3)$$

But clearly c_t satisfies $c_t > d_{s_{j-1}}$, by using the hypothesis that $c_t > c_{s_j}$ and (2). Thus, we must have that $c_t \geq c_{s_{j-1}}$, if c_t is not to satisfy (3). Now, since $c_{s_{j-1}}$ is positive by definition, we get that so is c_t . Thus, in fact, $c_t > c_{s_{j-1}}$. \square

Lemma 6.10. *Let t be a positive integer such that $s_j < t \neq s_{j+1}$ for some $1 \leq j \leq k-1$. Then exactly one of the following holds.*

- (1) $c_t > c_{s_1}$.
- (2) $c_t < c_{s_{j+1}}$.

Proof. Notice that if $c_t > c_{s_j}$, then Lemma 6.9 implies that $c_t > c_{s_1}$, which is precisely the first condition in the claim.

Now, assume that $c_{s_j} > c_t > c_{s_{j+1}}$. The definition of s_{j+1} implies that $c_{s_j} > c_{s_{j+1}} \geq d_{s_j}$. Thus we get that

$$c_{s_j} > c_t \geq d_{s_j}. \quad (4)$$

But since $c_t > c_{s_{j+1}}$ and $t > s_j$, (4) and the recursive definition of the elements of S would imply that s_{j+1} equals t . This is clearly not the case. Therefore $c_t \leq c_{s_{j+1}}$. Again using the fact that $c_{s_{j+1}}$ is positive, we get that $c_t < c_{s_{j+1}}$. \square

This brings us to the following important lemma which states that c_{s_k} occupies the rightmost box in the bottommost row of size i in α , where recall that α is the shape of τ . Here we are identifying α with its reverse composition diagram. Establishing that c_{s_k} occupies the rightmost box in a row of size i is the same as proving that $d_{s_k} = 0$. That it belongs to the bottommost row of length i requires us to prove that an entry c_t that lies strictly south of c_{s_k} is either 0 or is such that its neighbour $d_t > 0$.

Lemma 6.11. (1) $d_{s_k} = 0$.
 (2) If $t > s_k$ and $c_t > 0$, then $c_t > c_{s_1}$ and $d_t > 0$.

Proof. Consider first the i -th and $i+1$ -th columns of T_\square . For any positive integer r satisfying $1 \leq r \leq \min$, consider the following multisets.

$$\begin{aligned} X_r &= \{a_j : a_{\min} \geq a_j \geq b_r\} \\ Y_r &= \{b_j : b_{\min} \geq b_j \geq b_r\} \end{aligned}$$

Notice that X_r and Y_r can be multisets only when $b_r = 0$. Notice further that if $b_r = 0$, then the cardinalities of the multisets X_r and Y_r are equal. For a generic r , the following diagram describes the elements of X_r (boxes shaded purple in the i -th column) and Y_r (boxes shaded orange in the $i+1$ -th column).

a_1	b_1
\vdots	\vdots
a_j	b_j
\vdots	\vdots
a_r	b_r
\vdots	\vdots
a_{\min}	b_{\min}
$a_{\min+1}$	$b_{\min+1}$
\vdots	\vdots
a_n	b_n

Assume, contrary to what is to be proved, that $d_{s_k} > 0$. Then, we know that there exists a unique positive integer r such that $1 \leq r \leq n$ and $d_{s_k} = b_r$, and hence $b_r > 0$. Note also that $c_{s_k} \leq c_{s_1} < b_{min+1}$. This follows since $c_{s_1} = a_{min}$ and $a_{min} < b_{min+1}$.

Now, since $c_{s_k} \geq d_{s_k}$, we have that

$$b_r = d_{s_k} < b_{min+1}.$$

Using the above, combined with the fact that b_1 is always 0, we get that $1 < r \leq min$. Thus, we can consider the multisets X_r and Y_r . In fact, they are sets consisting of positive integers as $b_r > 0$. We have, by the definition of Y_r , that

$$d_{s_k} = b_r \in Y_r.$$

Furthermore, since

$$a_{min} = c_{s_1} \geq c_{s_k} \geq d_{s_k} = b_r,$$

we have $c_{s_k} \in X_r$. Notice now the important fact that the elements of Y_r can only be neighbours of elements in X_r , when we consider their respective positions in the i -th and $i + 1$ -th columns of τ_{\square} . This follows from Lemma 6.7, in a manner similar to the remark following it.

We know that $a_{r-1} \geq b_r = d_{s_k}$ as $r \leq min$. Hence, we obtain the following inequality.

$$|X_r| \geq |Y_r| + 1$$

This in turn implies

$$|X_r \setminus \{c_{s_k}\}| \geq |Y_r \setminus \{d_{s_k}\}| + 1. \quad (5)$$

Now consider the i -th and $i + 1$ -th columns of τ_{\square} . Each element of Y_r neighbours some element of X_r . Hence each element of $Y_r \setminus \{d_{s_k}\}$ neighbours a unique element of $X_r \setminus \{c_{s_k}\}$. Now, (5) implies that there is at least one element of $X_r \setminus \{c_{s_k}\}$ whose neighbour does not belong to $Y_r \setminus \{d_{s_k}\}$. This neighbour has to be strictly smaller than d_{s_k} clearly, as $d_{s_k} = b_r$ is the smallest entry in Y_r . The strictness follows from $d_{s_k} > 0$.

If, in τ_{\square} , all elements of the set $X_r \setminus \{c_{s_k}\}$ lie strictly north of the box occupied by c_{s_k} , then the argument earlier implies that for some $c_j \in X_r \setminus \{c_{s_k}\}$, we have $d_j < d_{s_k}$. But this corresponds to a triple rule violation because of the existence of the configuration shown below.

$$\begin{array}{c} \begin{array}{|c|c|} \hline c_j & d_j \\ \hline \end{array} \\ \vdots \\ \begin{array}{|c|} \hline d_{s_k} \\ \hline \end{array} \end{array} \quad \text{satisfying } c_j \geq d_{s_k} > d_j \geq 0$$

Hence, there is at least one element $c_j \in X_r \setminus \{c_{s_k}\}$ that lies strictly south of the box occupied by c_{s_k} . Any such element has to be greater than c_{s_k} . If not then it would satisfy the inequality $c_{s_k} > c_j \geq d_{s_k}$. But this would allow us to define s_{k+1} , which we know does not exist. Hence $c_j > c_{s_k}$. But then Lemma 6.9 implies that $c_j > c_{s_1}$. As $c_{s_1} = a_{min}$, this contradicts the fact

that $c_j \in X_r \setminus \{c_{s_k}\}$. Hence, we conclude that $d_{s_k} = 0$. This establishes the first part of the claim.

As for the second part, note for any positive integer $j > s_k$ such that $c_j > 0$, we must have $c_j > c_{s_k}$ (again by the fact that s_{k+1} does not exist). Lemma 6.9 implies that $c_j > c_{s_1}$, while Lemma 6.7 implies that $d_j \geq b_{\min+1} > 0$. This establishes the second part of the claim. \square

Remark 6.12. *The lemma above implies that position of c_{s_k} in τ only depends on the shape of τ , and not the filling itself.*

We would like to be able to construct the integers s_j in reverse, since locating c_{s_k} is easier than locating c_{s_1} in τ_\square . Thus, we would like an algorithm that takes as input the integer s_k (and τ_\square) and terminates by giving s_1 as output. Such an algorithm would give us a way of computing $f_{i+1}(T) = a_{\min} = c_{s_1}$ given τ . Before we achieve this aim, we need some notation.

Let q_1 be the greatest positive integer such that $d_{q_1} = 0$ and $c_{q_1} > 0$. Define a sequence of positive integers q_j satisfying $1 \leq q_j \leq n$ for $j \geq 2$ recursively according to the following conditions.

- (1) $q_j < q_{j-1}$,
- (2) q_j is the greatest positive integer such that

$$c_{q_j} > c_{q_{j-1}} \geq d_{q_j}.$$

By virtue of Lemma 6.11, we have that $q_1 = s_k$. In fact, $q_j = s_{k+1-j}$ for $1 \leq j \leq k$ as the next lemma shows.

Lemma 6.13. *For $1 \leq j \leq k$, we have $q_j = s_{k+1-j}$.*

Proof. Suppose that for some $1 \leq j \leq k-1$, we know that $q_j = s_{k+1-j}$. We will show that $q_{j+1} = s_{k-j}$. By the definition of q_{j+1} , we have

$$c_{q_{j+1}} > c_{q_j} \geq d_{q_{j+1}}. \quad (6)$$

Furthermore, since $q_j = s_{k+1-j}$, we have

$$c_{s_{k-j}} > c_{q_j} \geq d_{s_{k-j}}. \quad (7)$$

The recursive definition of q_{j+1} combined with (6) and (7) implies $q_{j+1} \geq s_{k-j}$. Suppose, contrary to what we seek to prove, that $q_{j+1} > s_{k-j}$. Thus, we have that $q_j > q_{j+1} > s_{k-j}$. Diagrammatically, this corresponds to the following configuration in τ_\square .

$$\begin{array}{c} \begin{array}{|c|c|} \hline c_{s_{k-j}} & d_{s_{k-j}} \\ \hline \end{array} \\ \vdots \\ \begin{array}{|c|c|} \hline c_{q_{j+1}} & d_{q_{j+1}} \\ \hline \end{array} \\ \vdots \\ \begin{array}{|c|} \hline c_{q_j} \\ \hline \end{array} \end{array}$$

From (6) and (7), we have $c_{q_{j+1}} > d_{s_{k-j}}$. Since $q_j = s_{k+1-j}$, we must have $c_{q_{j+1}} > c_{s_{k-j}}$. To see this, notice that had $c_{q_{j+1}} < c_{s_{k-j}}$ been true, the recursive definition of s_{k+1-j} given s_{k-j} would have implied that $s_{k+1-j} \leq q_{j+1}$, which is clearly not the case. Thus, we get that $c_{q_{j+1}} > c_{s_{k-j}}$.

But then, Lemma 6.9 implies $c_{q_{j+1}} > c_{s_1}$. By Lemma 6.7, this implies that

$$d_{q_{j+1}} \geq b_{\min+1}. \quad (8)$$

Since $c_{s_1} \geq c_{s_{k+1-j}} = c_{q_j}$, we know that $c_{q_j} < b_{\min+1}$, as $c_{s_1} = a_{\min} < b_{\min+1}$. But then the inequality in (8) is in contradiction to the inequality in (6). Hence $q_{j+1} = s_{k-j}$. Since $q_1 = s_k$, the claim follows by induction. \square

Thus, now we know that $q_k = s_1$. The next lemma will show that q_{k+1} does not exist.

Lemma 6.14. *There does not exist a positive integer t such that $t < s_1$ and*

$$c_t > c_{s_1} \geq d_t.$$

Proof. Suppose such a t exists. By Lemma 6.7, we have $d_t \geq b_{\min+1}$. But we also know that $b_{\min+1} > a_{\min} = c_{s_1}$. Thus, the inequality in the statement of the lemma can not be satisfied. \square

We now know how to compute c_{s_1} given τ_{\square} . Our next aim is to remove c_{s_1} from the i -th column of τ_{\square} and then rearrange entries therein so as to obtain $(\rho^{-1}(\text{tjdt}_{i+1}(T)))_{\square}$. The short algorithm that accomplishes this is presented below.

Algorithm 6.15. (1) For $j = k, k-1, \dots, 2$ in that order, place $c_{q_{j-1}}$ in the box that originally contained c_{q_j} .
 (2) Place a 0 in the box that originally contained c_{q_1} . Call the resulting tableau $\beta(\tau_{\square})$.
 (3) If there is a row in $\beta(\tau_{\square})$ comprising entirely of 0s, we remove that row. The resulting rectangular filling is $(\rho^{-1}(\text{tjdt}_{i+1}(T)))_{\square}$. If we omit 0s altogether, we obtain what we will call $\phi_{i+1}(\tau)$.

We will demonstrate the above algorithm with an example. The proof of validity of the algorithm will be presented subsequently.

Example 6.16. Suppose that

$$T_{\square} = \begin{array}{|c|c|c|c|c|} \hline 7 & 3 & 0 & 0 & 0 \\ \hline 8 & 5 & 0 & 0 & 0 \\ \hline 9 & 6 & 2 & 1 & 0 \\ \hline \end{array}.$$

Say we want to compute $\text{tjdt}_3(T)$. It is clear that if we start a backward jeu de taquin slide from the addable node in the third column, then 6 is the first entry that moves horizontally. Thus $\text{tjdt}_3(T)$ would be obtained by removing 6 from the second column. Then, we have

$$(\text{tjdt}_3(T))_{\square} = \begin{array}{|c|c|c|c|c|} \hline 7 & 0 & 0 & 0 & 0 \\ \hline 8 & 3 & 0 & 0 & 0 \\ \hline 9 & 5 & 2 & 1 & 0 \\ \hline \end{array}.$$

Now we try and repeat the above with the SSRCT $\tau = \rho^{-1}(T)$ using Algorithm 6.15. We have

$$\tau_{\square} = \begin{array}{|c|c|c|c|c|} \hline 7 & 6 & 2 & 1 & 0 \\ \hline 8 & 5 & 0 & 0 & 0 \\ \hline 9 & 3 & 0 & 0 & 0 \\ \hline \end{array}.$$

Then we have $q_1 = 3$ as c_3 is the lowermost non-zero entry in the second column that has a neighbour equalling 0. Since $c_2 > c_3 \geq d_2$ we have $q_2 = 2$. Similarly, we have $q_3 = 1$. Thus,

$$\beta(\tau_{\square}) = \begin{array}{|c|c|c|c|c|} \hline 7 & 5 & 2 & 1 & 0 \\ \hline 8 & 3 & 0 & 0 & 0 \\ \hline 9 & 0 & 0 & 0 & 0 \\ \hline \end{array}.$$

As there is no row comprising entirely of 0s, by the third step in the algorithm, we have that $(\rho^{-1}(\text{tjdt}_3(T)))_{\square}$ is the tableau above. We can remove the 0s to get a better picture. Then Algorithm 6.15 implies that

$$\phi_3(\tau) = \begin{array}{|c|c|c|c|} \hline 7 & 5 & 2 & 1 \\ \hline 8 & 3 & & \\ \hline 9 & & & \\ \hline \end{array}.$$

We also have

$$\text{tjdt}_3(T) = \begin{array}{|c|c|c|c|} \hline 7 & & & \\ \hline 8 & 3 & & \\ \hline 9 & 5 & 2 & 1 \\ \hline \end{array}.$$

It is easily checked that $\phi_3(\tau) = \rho^{-1}(\text{tjdt}_3(T))$.

Now we will proceed to prove the validity of Algorithm 6.15 by showing that $\phi_{i+1}(\tau)$ is an SSRCT. Since $\phi_{i+1}(\tau)$ is obtained from $\beta(\tau_{\square})$ by removing extraneous 0s, we will work with $\beta(\tau_{\square})$ as it is more convenient. We need to show that $\beta(\tau_{\square})$ satisfies the conditions below.

- Excepting 0s, the first column of $\beta(\tau_{\square})$ is strictly decreasing from bottom to top.
- It has no triple rule violations.
- The rows are weakly decreasing from left to right.

The third condition above is verified by an easy argument: Note that if the position of an entry in the i -th column remains unchanged, then the condition is automatically verified. Otherwise the entry c_{s_j} is replaced by $c_{s_{j+1}}$ for some $1 \leq j \leq k-1$. Since $c_{s_{j+1}} \geq d_{s_j}$ and $c_{s_j} > c_{s_{j+1}}$, we see that the algorithm does not change the fact that the rows were weakly decreasing from left to right. Finally, notice that c_{s_k} is replaced by a 0, but since d_{s_k} was already 0 by Lemma 6.11, we still have rows weakly decreasing from left to right.

Now we will verify that the first condition holds as well. Notice that if $i \geq 2$, then no changes are ever made to the first column by Algorithm 6.15. Hence, in this scenario $\phi_{i+1}(\tau)$ has the entries in the first column strictly decreasing from bottom to top. Consider now the case $i = 1$. Then, it is easily seen that Algorithm 6.15 replaces the bottom most non-zero

entry (in the first column) whose neighbour is a 0 with a 0, and no other change is made. Hence, if one omits the 0s, the first column of $\beta(\tau_\square)$ is strictly decreasing from bottom to top.

Thus, we only need to prove that there are no triple rule violations in $\beta(\tau_\square)$. We will do this in cases.

Lemma 6.17. *In $\beta(\tau_\square)$, consider any configuration of the type shown below*

$$\begin{array}{|c|c|} \hline x & y \\ \hline \vdots & \\ \hline z & \\ \hline \end{array},$$

where y and z belong to the $i + 1$ -th column. If $z > 0$ and $z > y$, then $z > x$.

Proof. Notice that Algorithm 6.15 does not alter the position of any entry in the $i + 1$ -th column. Let x' be the entry in τ_\square in the box currently occupied by x in $\beta(\tau_\square)$. Since there are no triple rule violations in τ_\square , we know that

$$z > 0 \text{ and } z > y \implies z > x' \quad (9)$$

We have two possibilities to deal with. Firstly, if $x = x'$, then the claim follows easily from (9). Now, assume $x \neq x'$. Algorithm 6.15 implies that $x < x'$. Again, (9) implies the claim. \square

Lemma 6.18. *In $\beta(\tau_\square)$, consider any configuration of the type shown below*

$$\begin{array}{|c|c|} \hline x & y \\ \hline \vdots & \\ \hline z & \\ \hline \end{array},$$

where y and z belong to the i -th column. If $z > 0$ and $z > y$, then $z > x$.

Proof. Firstly, notice that we only need to worry about this configuration when $i \geq 2$.

Let x' , y' and z' be the entries in τ_\square in the boxes corresponding to those occupied by x , y and z in $\beta(\tau_\square)$ respectively. Since Algorithm 6.15 does not change the $i - 1$ -th column, we have $x = x'$. This along with the fact that there are no triple rule violations in τ_\square implies

$$z' > 0 \text{ and } z' > y' \implies z' > x.$$

If $y = y'$ and $z = z'$, then we have nothing to prove.

Now assume that $z = z'$ but $y \neq y'$. Then we have two cases. In the first case, we have $y' = c_{s_k}$ and thus, $y = 0$. Furthermore, since z occupies a box below c_{s_k} , we know that $z > c_{s_1}$, by Lemma 6.11. Now the claim follows readily for this case. In the second case, we have $y' = c_{s_j}$ and thus, $y = c_{s_{j+1}}$ for some $1 \leq j \leq k - 1$. The possible relative positions of

y , y' and z in τ_\square are shown below.

$$\begin{array}{cc} \boxed{y'} & \boxed{y'} \\ \vdots & \vdots \\ \boxed{y} & \boxed{z} \\ \vdots & \vdots \\ \boxed{z} & \boxed{y} \end{array}$$

According to Lemma 6.10, we have either $z > c_{s_1}$ (and hence $z > y, y'$) or $z < y$ (and hence $z < y'$). In any case, we see that if $z > 0$ and $z > y$, then $z > x$.

Now assume that $y \neq y'$ and $z \neq z'$. Then we must have $y = c_{s_{i_1}}$ and $z = c_{s_{i_2}}$ where $2 \leq i_1 < i_2 \leq k$. But then the situation where $z > y$ does not arise. \square

Thus, we have established that $\phi_{i+1}(\tau)$ is an SSRCT indeed. We will prove another lemma that is of similar flavour to the previous two lemmas, and which we will soon use.

Lemma 6.19. *In $\beta(\tau_\square)$, consider a configuration of the type*

$$\boxed{x \mid y} \quad \text{where } x \text{ is in } i\text{-th column and } y \text{ in } i+1\text{-th column.}$$

If $c_{s_1} > y \geq 0$ then $c_{s_1} > x$.

Proof. For any configuration of the type in the statement of the claim, we first show that c_{s_1} can neither equal x nor y .

Recall that $c_{s_1} = a_{\min}$ is the first entry that moves horizontally when computing $\text{jdt}_{i+1}(T)$. This entry is removed from the i -th column of τ_\square when $\beta(\tau_\square)$ is computed. Hence it can not equal x . Lemma 6.6 implies it can not equal y either.

If $c_{s_1} > y \geq 0$, then we know that $y \leq b_{\min}$. But then Lemma 6.7 yields that $x < c_{s_1}$, which is what we wanted. \square

We are now in a position to state our first important theorem.

Theorem 6.20. *The following diagram commutes.*

$$\begin{array}{ccc} \text{SSRTs} & \xrightarrow{\rho^{-1}} & \text{SSRCTs} \\ \text{tjdt}_{i+1} \downarrow & & \downarrow \phi_{i+1} \\ \text{SSRTs} & \xrightarrow{\rho^{-1}} & \text{SSRCTs} \end{array}$$

Proof. We have already established that $\phi_{i+1}(\tau)$ is an SSRCT. This observation and the fact that we have removed the same entry from the i -th column in both T and τ while ensuring that none of the other entries have changed columns, establishes the claim. \square

We define another SSRCT using the procedure ϕ_i as follows.

$$\Phi_i(\tau) = \begin{cases} \phi_2 \circ \phi_3 \circ \cdots \circ \phi_i(\tau) & i > 1 \\ \tau & i = 1 \end{cases}$$

Let $\gamma = \text{sh}(\Phi_i(\tau)) = (\gamma_1, \dots, \gamma_s)$. Let $\gamma' = (\gamma, i)$ be the composition obtained by appending a part of length i to γ . Consider the filling of shape $\gamma' // (1)$ where the first s rows are filled as $\Phi_i(\tau)$. Fill the last row with the entries of $H_i(T)$ in decreasing order from left to right. Call the resulting filling $\mu_i(\tau)$.

Theorem 6.21. *With the notation as above, the following diagram commutes.*

$$\begin{array}{ccc} \text{SSRTs} & \xrightarrow{\rho^{-1}} & \text{SSRCTs} \\ \text{jdt}_i \downarrow & & \downarrow \mu_i \\ \text{SSRTs} & \xrightarrow{\rho_{(1)}^{-1}} & \text{SSRCTs} \end{array}$$

Proof. First, we start by showing that $\mu_i(\tau)$ is actually an SSRCT. Notice that the entries in every column are all distinct, using Lemma 6.6. Next, since the rows are weakly decreasing by construction, and the top s rows already form an SSRCT, we just need to check that there is no configuration of the form

$$\begin{array}{c} \boxed{x} \quad \boxed{y} \\ \vdots \\ \boxed{z} \end{array} \quad \text{satisfying } x \geq z > y \text{ and } z \text{ is in the bottommost row.}$$

But this follows from Lemma 6.19. Hence $\mu_i(\tau)$ is an SSRCT. Now the claim follows as the set of entries in each column of $\text{jdt}_i(T)$ and $\mu_i(\tau)$ is the same. \square

Given the theorem above, we are justified in saying that μ is an analogue of the backward jeu de taquin slide. From this point on, by a backward jeu de taquin slide on SSRCTs of straight shape we will mean an application of μ .

6.1. Skew shape after μ . If we perform backward jeu de taquin slides on SSRTs of straight shape, it is easy to predict the resulting shape once the slide is completed. In the case of SSRCTs, when we apply the procedure μ , it is not immediate what the resulting shape is. The aim of this subsection is to use the operators on compositions defined in Section 4 to predict it.

Lemma 6.22. *The shape of $\phi_{i+1}(\tau)$ is $\mathfrak{d}_i(\alpha)$ if $i \geq 1$.*

Proof. It is clear from Algorithm 6.15 that box that contained c_{s_k} in τ_{\square} contains 0 in $\phi_{i+1}(\tau_{\square})$. We also know from Lemma 6.11 that all parts of α that are below α_{s_k} in the reverse composition diagram corresponding to α have either length strictly less than α_{s_k} or strictly greater than α_{s_k} . Furthermore, the same lemma also implies that α_{s_k} has length i . Thus, the claim follows. \square

As an immediate corollary, we obtain the following.

Corollary 6.23. *The shape of $\Phi_{i+1}(\tau)$ is $\mathfrak{v}_{[i]}(\alpha)$ if $i \geq 0$.*

Proof. Φ_1 fixes the SSRCT τ . In this case the claim is clear. If we are computing $\Phi_{i+1}(\tau)$ for $i \geq 1$, we are computing $\phi_2 \circ \cdots \circ \phi_{i+1}(\tau)$. Using Lemma 6.22 repeatedly, we get that $\Phi_{i+1}(\tau)$ has shape $\mathfrak{d}_1 \cdots \mathfrak{d}_i(\alpha) = \mathfrak{v}_{[i]}(\alpha)$. \square

Theorem 6.24. *The SSRCT $\mu_i(\tau)$ is of shape $\gamma \parallel (1)$ where $\gamma = u_i(\alpha)$.*

Proof. This follows from the description of the procedure μ_i and Corollary 6.23. \square

7. BACKWARD JEU DE TAQUIN SLIDES FOR SSRCTS OF SKEW SHAPE

After giving a description for the backward jeu de taquin slide for SSRCTs of straight shape, it is only natural to ask for a generalization, that is, a backward jeu de taquin slide for SSRCTs of other shapes. For the proofs in this section, we will consider SSRCTs whose entries are drawn from the ordered alphabet $X = A \cup \bar{A}$ where

$$\begin{aligned} A &= \{1 < 2 < 3 < \cdots\} \\ \bar{A} &= \{\bar{1} < \bar{2} < \bar{3} < \cdots\}. \end{aligned}$$

We will further assume that $\bar{1}$ is greater than any letter in A . Thus, we have a total order on X .

Consider a skew reverse composition shape $\alpha \parallel \beta$. Let τ_o be an SSRCT of shape $\alpha \parallel \beta$ constructed using the alphabet A , and let $\bar{\tau}$ be an SSRCT of straight shape β constructed using the alphabet \bar{A} . Then we will denote by τ the SSRCT of straight shape α formed by filling the inner shape in τ_o according to $\bar{\tau}$.

Let $i \geq 1$. Assume, for the moment, that we are computing $\phi_{i+1}(\tau)$. We will show that the shape of the SSRCT obtained by considering entries that belong to A in $\phi_{i+1}(\tau)$ is independent of the filling $\bar{\tau}$. The example next will clarify what we are aspiring to.

Example 7.1. *Consider the two tableaux below.*

$$\tau_o = \begin{array}{|c|c|c|c|} \hline 3 & & & \\ \hline \bullet & \bullet & 2 & 1 \\ \hline \bullet & \bullet & \bullet & 5 \\ \hline \bullet & 4 & & \\ \hline \end{array} \text{ and } \bar{\tau} = \begin{array}{|c|c|c|} \hline \bar{3} & \bar{1} & \\ \hline \bar{5} & \bar{4} & \bar{2} \\ \hline \bar{6} & & \\ \hline \end{array}$$

Thus

$$\tau = \begin{array}{|c|c|c|c|} \hline 3 & & & \\ \hline \bar{3} & \bar{1} & 2 & 1 \\ \hline \bar{5} & \bar{4} & \bar{2} & 5 \\ \hline \bar{6} & 4 & & \\ \hline \end{array}$$

Then $\phi_3(\tau)$ is the SSRCT below.

3			
$\bar{3}$	4	2	1
$\bar{5}$	$\bar{4}$	$\bar{2}$	5
$\bar{6}$			

If we replace the entries in the above tableau that belong to \bar{A} by bullets, then we obtain the following SSRCT.

3			
•	4	2	1
•	•	•	5
•			

We would like to demonstrate that the above tableau is obtained independent of our choice of the filling $\bar{\tau}$.

We will use the same notation as the one we established when we described the process ϕ_{i+1} . Thus, our focus is on the i -th and $i+1$ -th columns of T_{\square} and τ_{\square} respectively, as shown below. Recall that $\rho(\tau) = T$.

a_1	b_1	c_1	d_1
a_2	b_2	c_2	d_2
\vdots	\vdots	\vdots	\vdots
a_n	b_n	c_n	d_n

The only difference from the earlier situation is that the entries are now allowed to be elements of \bar{A} . We will recall the notation here for the sake of convenience. We have that a_{min} is the first entry that moves horizontally when a backward jeu de taquin slide is initiated from the $i+1$ -th column in T . Hence min is the smallest positive integer such that $a_{min} < b_{min+1}$. Also we assume that, in τ_{\square} , the entry corresponding to a_{min} is c_{s_1} , that is, s_1 is the positive integer such that $c_{s_1} = a_{min}$. Finally, we define recursively integers s_j satisfying $1 \leq s_j \leq n$ for $j \geq 2$ using the following criteria.

- (1) $s_j > s_{j-1}$,
- (2) c_{s_j} is the greatest positive integer that satisfies

$$c_{s_{j-1}} > c_{s_j} \geq d_{s_{j-1}}.$$

Clearly, only finitely many s_j exist. Let the maximum be given by s_k and let

$$S = \{s_1, \dots, s_k\}.$$

Lemma 7.2. *Assume $c_{s_1} \in \bar{A}$. Let $1 \leq j \leq k$ be the greatest positive integer such that c_{s_j} belongs to \bar{A} . If $t > s_j$ is a positive integer such that $c_t > 0$, then one of the following statements holds.*

- (1) $c_t \in A$.
- (2) $c_t \in \bar{A}$ and $d_t \in \bar{A}$.

Proof. Firstly, note that $c_{s_1} \in \bar{A}$ implies that $b_{\min+1} \in \bar{A}$. Notice also that if $c_{s_j} \in \bar{A}$, then for all $1 \leq r \leq j$, we must have that $c_{s_r} \in \bar{A}$ as $c_{s_1} > \dots > c_{s_j} > \dots > c_{s_k}$.

If $c_t \in A$, then the first condition is already satisfied. Hence assume that $c_t \in \bar{A}$. If $j = k$, then Lemma 6.11 implies that $c_t > c_{s_1}$. Then, Lemma 6.7 implies $d_t \geq b_{\min+1}$. Hence $d_t \in \bar{A}$.

Now assume that $j < k$. We have that $c_{s_j} \in \bar{A}$ and $c_{s_{j+1}} \in A$. Since the definition of s_{j+1} implies that

$$c_{s_j} > c_{s_{j+1}} \geq d_{s_j},$$

we get that $d_{s_j} \in A$.

By our assumptions, we know that $t > s_j$ and $c_t \in \bar{A}$. Using this and the fact that j is the largest integer such that $c_{s_j} \in \bar{A}$, we conclude that $t \neq s_{j+1}$. But then, we must have that $c_t > c_{s_j}$. To see this, note that if $c_t < c_{s_j}$, then we have the following inequality using the fact that $d_{s_j} \in A$.

$$c_{s_j} > c_t > d_{s_j}$$

But then, we get that $c_{s_{j+1}} > c_t$ and hence, must belong to \bar{A} , which is not the case. Hence $c_t > c_{s_j}$ indeed.

Lemma 6.10 implies the inequality $c_t > c_{s_1}$ and then Lemma 6.7 implies that $d_t \geq b_{\min+1}$. Thus, we get that $d_t \in \bar{A}$. \square

Note that the above lemma assumes nothing about the filling $\bar{\tau}$.

Lemma 7.3. *Assume $c_{s_1} \in \bar{A}$. Consider the SSRCT of straight shape inside $\phi_{i+1}(\tau)$ formed by the entries that belong to \bar{A} . The shape of this SSRCT is $\mathfrak{d}_i(\beta)$.*

Proof. Let $1 \leq j \leq k$ be the greatest positive integer such that $c_{s_j} \in \bar{A}$. Then from Lemma 7.2, it follows that c_{s_j} occupies the rightmost box in the bottommost row of length i in the reverse composition diagram of β . According to Algorithm 6.15, this entry gets replaced either by an element of A or 0. In either case, the shape of the SSRCT formed by the entries that belong to \bar{A} is $\mathfrak{d}_i(\beta)$. \square

Lemma 7.4. *Assume that $c_{s_1} \in A$. Then the SSRCT in $\phi_{i+1}(\tau)$ formed by considering the entries that belong to A is independent of the choice of $\bar{\tau}$. Furthermore, this SSRCT has shape $\mathfrak{d}_i(\alpha) \parallel \beta$.*

Proof. Since c_{s_1} belongs to A and $c_{s_1} > c_{s_j}$ for all $2 \leq j \leq k$, we get that $c_{s_j} \in A$ for $2 \leq j \leq k$. Note by Lemma 6.11 that c_{s_k} occupies the rightmost box in the bottommost part of length i in the reverse composition diagram of α . Now Algorithm 6.15 immediately implies the claim. \square

Lemma 7.5. *Assume that $c_{s_1} \in \bar{A}$. Then the SSRCT in $\phi_{i+1}(\tau)$ formed by considering the entries that belong to A is independent of the choice of $\bar{\tau}$. Furthermore, this SSRCT has shape $\mathfrak{d}_i(\alpha) \parallel \mathfrak{d}_i(\beta)$.*

Proof. Let j be the greatest positive integer satisfying $1 \leq j \leq k$ such that $c_{s_j} \in \bar{A}$. By Lemma 7.3, we know that c_{s_j} occupies the rightmost box in the bottommost row of length i in the reverse composition diagram of β . Also, by Lemma 6.11 we know that c_{s_k} occupies the rightmost box in the bottommost row of length i in the reverse composition diagram of α (recall also that we draw the reverse composition diagram of β in the bottom left corner of the reverse composition diagram of α when depicting $\alpha \parallel \beta$). Observe now that s_j does not depend on $\bar{\tau}$. Thus, from Algorithm 6.15, it follows that the SSRCT in $\phi_{i+1}(\tau)$ formed by the entries that belong to A is independent of $\bar{\tau}$ and that it has shape $\mathfrak{d}_i(\alpha) \parallel \mathfrak{d}_i(\beta)$. \square

Now we define $\mu_i(\tau_o)$ as follows.

$\mu_i(\tau_o) =$ the SSRCT formed by considering the entries that belong to A in $\mu_i(\tau)$.

It is not clear from the above definition that $\mu_i(\tau_o)$ does not depend on the choice of $\bar{\tau}$. We establish this next.

Let $T_o = \rho_\beta(\tau_o)$ and $\bar{T} = \rho(\bar{\tau})$. Define T to be $\rho(\tau)$. Consider a backward jeu de taquin slide on T starting from an addable node in column i . Let j be the greatest integer satisfying $1 \leq j \leq i - 1$ such that entry that moves horizontally from column j to column $j + 1$ when computing $\text{jdt}_i(T)$ is an element of \bar{A} . If there is no such j , define j to be 0. If $j \geq 1$, then the entries moving horizontally when computing $\text{jdt}_i(T)$ from all columns between 1 and j (inclusive) are all elements of \bar{A} , because the entries moving horizontally during a backward jeu de taquin slide increase weakly as we move from column $i - 1$ to column 1. Now by using Theorem 6.20, we get that if we compute $\phi_2 \circ \dots \circ \phi_i(\tau)$, all the entries that exit from columns between 1 and j belong to \bar{A} .

The preceding discussion along with Lemmas 7.4 and 7.5 implies the following proposition.

Proposition 7.6. *Let γ be the SSRCT $\Phi_i(\tau)$ where $i \geq 1$. Then the SSRCT formed by considering the entries in γ that belong to \bar{A} has shape $\mathfrak{v}_{[j]}(\beta)$. The SSRCT formed by considering the entries in γ that belong to A has shape $\mathfrak{v}_{[i-1]}(\alpha) \parallel \mathfrak{v}_{[j]}(\beta)$. Furthermore, this SSRCT is independent of the choice of $\bar{\tau}$.*

This proposition along with our description of the procedure μ prior to Theorem 6.21, implies the following theorem.

Theorem 7.7. *Let γ be the SSRCT $\mu_i(\tau)$ where $i \geq 1$. Then the SSRCT formed by considering the entries in γ that belong to \bar{A} has shape $u_{j+1}(\beta) \parallel (1)$. The SSRCT formed by considering the entries in γ that belong to A has shape $u_i(\alpha) \parallel u_{j+1}(\beta)$, and is independent of the choice of $\bar{\tau}$.*

The theorem above proves that $\mu_i(\tau_o)$ is well-defined and does not depend on the choice of filling $\bar{\tau}$. From this point on, we will not require the alphabet \bar{A} .

8. A NEW POSET ON COMPOSITIONS

In this section we will introduce a new poset structure on the set of compositions that arises from performing backward jeu de taquin slides on SSRCTs. But before that we need to understand the link between maximal chains in Young's lattice \mathcal{Y} and SRTs.

8.1. Growth words and SRTs. With any maximal chain in \mathcal{Y} starting at the empty partition, we can associate a sequence of positive integers as explained next. Let

$$\lambda_0 = \emptyset \prec \lambda_1 \prec \cdots \prec \lambda_n$$

be a maximal chain. Then each partition λ_k for $1 \leq k \leq n$ is obtained by adding a box at an addable node of λ_{k-1} , say, in column i_k . The sequence $i_n \cdots i_1$ is called the *column growth word* corresponding to this maximal chain. Note that the column growth word is always *reverse lattice*, that is, in any suffix of the column growth word the number of j s weakly exceeds the number of $j+1$ s for all positive integers j . What we have outlined is clearly a bijection between reverse lattice words of length n and maximal chains in \mathcal{Y} starting at \emptyset and ending at a partition of n . We will now associate an SRT of size n given a column growth word $w = i_n \cdots i_1$. This is done by considering the corresponding maximal chain and assigning $n - k + 1$ to the box in λ_k that is not in λ_{k-1} . This SRT will be denoted by T_w .

Before we state our next lemma connecting the column reading word of T_w to its associated reverse lattice word, we require the notion of standardization. Given a word $w = i_n \cdots i_1$ where the i_j s are positive integers, define the *inversion set* as follows.

$$\text{Inv}(w) = \{(p, q) : i_p < i_q\}$$

There exists a unique permutation in $\sigma \in \mathfrak{S}_n$ such that $\text{Inv}(\sigma) = \text{Inv}(w)$. We call σ the *standardization* of w , and denote it by $\text{std}(w)$.

Lemma 8.1. *Let $w = i_n \cdots i_1$ be a reverse lattice word, and let $\sigma = \text{std}(w)$. Then the column reading word of T_w equals σ^{-1} .*

Proof. Notice that all the instances of a positive integer j in w are in positions given by the entries in the j -th column of T_w read in increasing order. This implies the claim. \square

Example 8.2. *Consider the reverse lattice word $w = 341123121$. Then we have that $\sigma = \text{std}(w) = 791258364$ in single line notation. Notice that T_w is the SSRT below.*

3			
4			
7	5	1	
9	8	6	2

The column reading word of T_w is the following permutation.

347958162

It can be checked that the permutation above is precisely σ^{-1} .

Now recall that if we perform our variant of the Robinson-Schensted algorithm on the column reading word of a tableau T , we recover T as the insertion tableau. This implies the following corollary to the previous lemma, establishing T_w as a insertion tableau for permutation associated naturally to it.

Corollary 8.3. *Let $w = i_n \cdots i_1$ be a reverse lattice word, and let $\sigma = \text{std}(w)$. Then $T_w = P(\sigma^{-1})$.*

This interpretation of T_w will be of importance to us in the next subsection.

8.2. A poset on compositions. We will now define a new poset on the set of compositions, denoted by \mathcal{R}_c , using its cover relation \prec_r defined by setting $\alpha \prec_r \beta$ if and only if $\beta = u_i(\alpha)$ for some $i \geq 1$. The order relation $<_r$ in \mathcal{R}_c is obtained by taking the transitive closure of the cover relation \prec_r .

Example 8.4. *Let $\alpha = (3, 1, 4, 2, 1)$ and $\beta = u_4(\alpha) = (2, 1, 4, 1, 4)$. Then $\alpha \prec_r \beta$ in \mathcal{R}_c .*

The next theorem is about counting maximal chains in \mathcal{R}_c , and the reader should compare it with the analogous result in the case of Young's lattice \mathcal{Y} [20, Proposition 7.10.3] and, more recently, in the case of \mathcal{L}_c [2, Proposition 2.11].

Theorem 8.5. *The number of maximal chains from \emptyset to a composition α in \mathcal{R}_c is equal to the number of SRCTs of shape α .*

To prove the above theorem, we will use our variant of the Robinson-Schensted algorithm defined in Subsection 2.5 and answer a more general question. We will start by proving a result that allows us to compute the resulting inner shape after a sequence of backward jeu de taquin slides has been applied to an SSRCT of straight shape.

Let τ be an SSRCT of shape α . Consider the computation of $\mu_{i_n} \cdots \mu_{i_1}(\tau)$, where we assume that the sequence of backward jeu de taquin slides we are executing is valid at all steps. This means that for all $1 \leq k \leq n-1$, the outer shape β underlying the SSRCT $\mu_{i_k} \cdots \mu_{i_1}(\tau)$ is such that $\tilde{\beta}$ has an addable node in column i_{k+1} . Furthermore, we also assume that $\tilde{\alpha}$ has an addable node in column i_1 , so that we can compute $\mu_{i_1}(\tau)$ to start with.

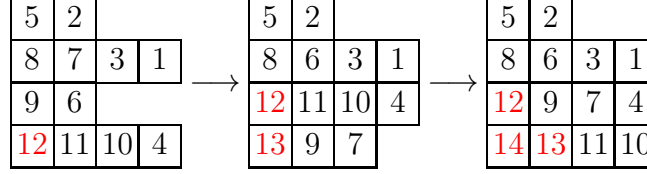
We will deviate a little from our earlier description of backward jeu de taquin slides for an SSRCT of straight shape. Let $\max(\tau)$ denote the maximum entry in the SSRCT τ (if $\tau = \emptyset$ then this is 0). We will assume that in computing $\mu_i(\tau)$, instead of getting a SSRCT of skew reverse composition shape, with the vacant lower left corner filled by a bullet \bullet , we insert the number $\max(\tau) + 1$ in the corner left vacant. In this way we ensure that $\mu_i(\tau)$ is also an SSRCT of straight shape. We will do backward jeu de taquin slides (the procedure jdt) on SSRTs of straight shape with a similar rule.

Example 8.6. Consider the SRCT $\tau =$

5	2
8	7
9	6
11	10

 of shape $(2, 4, 3, 2)$. We will compute $\mu_4 \mu_3 \mu_4(\tau)$ under the set of rules outlined prior to the example. Firstly, $\max(\tau) = 11$. We

obtain the following SSRCTs sequentially (where the entries that come in the lower left corner are highlighted).



We will next state a theorem that allows us to compute the final SRCT (under our new setup) obtained when a sequence of slides is applied to an empty SRCT \emptyset .

Theorem 8.7. *Let $w = i_n \cdots i_1$ be a reverse lattice word, and let $\sigma = \text{std}(w)$. Then*

$$\mu_{i_n} \cdots \mu_{i_1}(\emptyset) = \rho^{-1}(Q(\sigma)).$$

Proof. Let $\mu_{i_n} \cdots \mu_{i_1}(\emptyset) = \tau$. Clearly, τ is an SRCT. Now consider the corresponding sequence of slides starting from the empty SRT, also denoted \emptyset , and suppose that the final SRT obtained is T . Thus, $\text{jdt}_{i_n} \cdots \text{jdt}_{i_1}(\emptyset) = T$. From Theorem 6.21, it follows that $\rho(\tau) = T$ in this new setting.

Observe that in obtaining T , we encounter a sequence of SRTs for $1 \leq k \leq n$ as follows.

$$T_k = \text{jdt}_{i_k} \cdots \text{jdt}_{i_1}(\emptyset)$$

If we let λ_k denote $\text{sh}(T_k) \vdash k$, we get a maximal chain in \mathcal{Y} .

$$\emptyset \prec \lambda_1 \prec \cdots \prec \lambda_n$$

Thus, we have that $w = i_n \cdots i_1$ is the column growth word corresponding to the maximal chain above. This combined with the reversibility of jeu de taquin for SSRTs gives us the key fact that $T_w = e(T)$ (essentially as evacuation undoes backward jeu de taquin slides).

From Corollary 8.3 we know that $T_w = P(\sigma^{-1})$, and by Lemma 2.12, we know that $P(\sigma^{-1}) = e(Q(\sigma))$. Combining the previous two facts, we obtain the following.

$$e(T) = e(Q(\sigma))$$

Since evacuation is an involution as is shown in the appendix, we get that $T = Q(\sigma)$. This implies that the SRCT given by $\mu_{i_n} \cdots \mu_{i_1}(\emptyset) = \tau = \rho^{-1}(T)$, equals $\rho^{-1}(Q(\sigma))$. \square

We claim that the above theorem also supplies us with a bijection between the set of maximal chains from \emptyset to α in \mathcal{R}_c and the set of SRCTs of shape α .

Proof. (of Theorem 8.5) Consider a maximal chain in \mathcal{R}_c as shown below.

$$\emptyset <_r \alpha_1 <_r \cdots <_r \alpha_n = \alpha$$

Just as in the case of \mathcal{Y} , the maximal chain above gives us a reverse lattice word $w = i_n \cdots i_1$ where $\alpha_k = u_{i_k}(\alpha_{k-1})$ for $1 \leq k \leq n$. Associate the SRCT $\mu_{i_n} \cdots \mu_{i_1}(\emptyset)$ with this maximal chain. Note that the shape of this SRCT, by Theorem 6.24, is $u_{i_n} \cdots u_{i_1}(\emptyset)$ where \emptyset denotes the empty composition. But this is clearly α . To establish that the map associating the

SRCT $\mu_{i_n} \cdots \mu_{i_1}(\emptyset)$ to the reverse lattice word $i_n \cdots i_1$ obtained from the maximal chain is indeed a bijection, we will use the invertibility of jeu de taquin slides for SSRTs.

Consider an SRCT τ of shape α , and let $T = \rho(\tau)$. We will analyze the execution of the evacuation action on T . At every step a forward jeu de taquin slide is initiated from the lower left corner, and this slide finishes in an addable node to the shape underlying the rectified tableau. Noting down the columns of these addable nodes gives us a sequence of n positive integers i_n, i_{n-1}, \dots, i_1 . This sequence is **uniquely determined by T** (by the invertibility of jeu de taquin slides on SSRTs) and the word $w = i_n \cdots i_1$ is reverse lattice. Now since backward jeu de taquin slides undo the forward jeu de taquin slides, we get that

$$T = \text{jdt}_{i_n} \cdots \text{jdt}_{i_1}(\emptyset),$$

and hence that

$$\tau = \mu_{i_n} \cdots \mu_{i_1}(\emptyset).$$

Since τ had shape α to start with, we get that $u_{i_n} \cdots u_{i_1}(\emptyset) = \alpha$ as well (here again \emptyset refers to the empty composition). Now we can associate the following (unique) maximal chain from \emptyset to α in \mathcal{R}_c to τ .

$$\emptyset <_r u_{i_1}(\emptyset) <_r u_{i_2}u_{i_1}(\emptyset) <_r \cdots <_r u_{i_n} \cdots u_{i_1}(\emptyset) = \alpha.$$

This finishes the proof. □

We will give an example illustrating the ideas above.

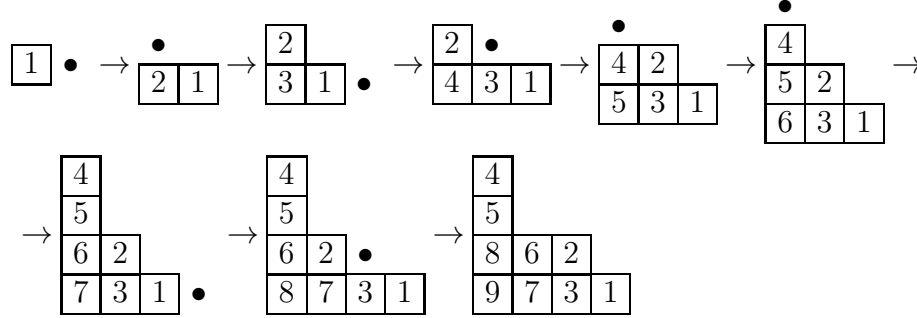
Example 8.8. To compute $\tau = \mu_3\mu_4\mu_1\mu_1\mu_2\mu_3\mu_1\mu_2\mu_1(\emptyset)$, note that the column growth word is 341123121 and its standardization (which is a permutation in \mathfrak{S}_9) in single line notation is $\sigma = 791258364$. On applying the Robinson-Schensted variant, σ maps to the following insertion tableau (left) and recording tableau (right).

$$P(\sigma) = \begin{array}{|c|c|c|c|} \hline 1 \\ \hline 2 \\ \hline 7 & 5 & 3 \\ \hline 9 & 8 & 6 & 4 \\ \hline \end{array} \quad Q(\sigma) = \begin{array}{|c|c|c|c|} \hline 4 \\ \hline 5 \\ \hline 8 & 6 & 2 \\ \hline 9 & 7 & 3 & 1 \\ \hline \end{array}$$

Then, by Theorem 8.7, we have that $\tau = \rho^{-1}(Q(\sigma))$.

$$\tau = \begin{array}{|c|c|c|c|} \hline 4 \\ \hline 5 \\ \hline 8 & 7 & 3 & 1 \\ \hline 9 & 6 & 2 & \\ \hline \end{array}$$

It is straightforward to check that $\text{jdt}_3\text{jdt}_4\text{jdt}_1\text{jdt}_1\text{jdt}_2\text{jdt}_3\text{jdt}_1\text{jdt}_2\text{jdt}_1(\emptyset)$ gives the following sequence of SRTs (the bullets indicating where the next slide begins).



Note that the final SRT obtained above is $\rho(\tau)$.

In the next subsection, we will use Theorem 8.7 and the proof strategy for Theorem 8.5 to consider the effect of a sequence of slides starting from a nonempty SRCT.

8.3. Inner shape after a sequence of slides. Let τ_1 be an SRCT of shape $\alpha \vdash n$, and let $T_1 = \rho(\tau_1)$ be of shape $\lambda \vdash n$. Considering the execution of the evacuation action on T_1 as in the proof of Theorem 8.5, we get a sequence of n positive integers i_n, i_{n-1}, \dots, i_1 . This sequence is uniquely determined by T_1 and the word $w = i_n \cdots i_1$ is reverse lattice.

Remark 8.9. Note that if we started with the final SRT in Example 8.8, and performed the procedure discussed prior to this remark, we will be obtain the same sequence of SRTs in reverse and the columns to which the bullets belong are precisely the integers i_n down to i_1 .

It follows that $T_1 = \text{jdt}_{i_n} \cdots \text{jdt}_{i_1}(\emptyset)$ which in turn implies that $\tau_1 = \mu_{i_n} \cdots \mu_{i_1}(\emptyset)$. Consider now the SRCT τ_2 defined below.

$$\tau_2 = \mu_{k_m} \cdots \mu_{k_1}(\tau_1)$$

Let $\rho(\tau_2)$ be T_2 . Clearly, we have that $T_2 = \text{jdt}_{k_m} \cdots \text{jdt}_{k_1}(T_1)$. Denote by u the word $k_m \cdots k_1$.

Let $v = u \cdot w$, where \cdot denotes concatenation of words, and let $\pi = \text{std}(v)$, $\sigma = \text{std}(w)$ and $\omega = \text{std}(u)$. Now, note that Theorem 8.7 implies that

$$\tau_2 = \rho^{-1}(Q(\pi)).$$

Observe additionally that the integers $m+n, m+n-1, \dots, n+1$ are the new entries added when computing $\mu_{k_m} \cdots \mu_{k_1}(\tau_1)$, and they form an SSRCT of size m (with all entries distinct) inside τ_2 . But this SSRCT is obtained by performing our insertion algorithm on the prefix of length m in π , and then applying ρ^{-1} to the partial Q -tableau obtained at this stage. If one subtracts n from all entries in this partial Q -tableau, one gets an SRT that is precisely the Q -tableau obtained after performing the insertion process on $\omega = \text{std}(u)$ (recall that standardization does not alter the Q -tableau).

From this point on, we switch perspectives again, and on performing backward jeu de taquin slides on SSRCTs, instead of filling the vacant box in the lower left corner with larger entries, we will fill them with bullets. The preceding discussion implies the following theorem.

Theorem 8.10. *Let τ_1 be an SRCT of straight shape α , and τ_2 be the SRCT $\mu_{j_m} \cdots \mu_{j_1}(\tau_1)$. Let $w = j_m \cdots j_1$ and $\omega = \text{std}(w) \in \mathfrak{S}_m$. Let $\beta = u_{j_m} \cdots u_{j_1}(\alpha)$ and $\gamma = \text{sh}(\rho^{-1}(Q(\omega)))$. Then τ_2 is an SRCT of shape $\beta // \gamma$.*

The following corollary is obtained from the theorem above.

Corollary 8.11. *Let τ_1 be an SRCT of straight shape α , and τ_2 be the SRCT $\mu_{j_m} \cdots \mu_{j_1}(\tau_1)$. Let $w = j_m \cdots j_1$ and $\omega = \text{std}(w) \in \mathfrak{S}_m$. Let $\beta = u_{j_m} \cdots u_{j_1}(\alpha)$. Then τ_2 is an SRCT of shape $\beta // (n)$ if and only if $j_1 < \cdots < j_m$, and it is an SRCT of shape $\beta // (1^n)$ if and only if $j_1 \geq \cdots \geq j_m$.*

SRT versions of the two results above also exist and they are also implied in the discussion earlier. We will recast the corollary above for SRTs as another corollary.

Corollary 8.12. *Let T_1 be an SRT of straight shape λ , and T_2 be the SRT $\text{jdt}_{j_m} \cdots \text{jdt}_{j_1}(T_1)$. Let $w = j_m \cdots j_1$ and $\omega = \text{std}(w) \in \mathfrak{S}_m$. Let $\beta = u_{j_m} \cdots u_{j_1}(\alpha)$ and $\delta = \tilde{\beta}$. Then T_2 is an SRT of shape $\delta / (n)$ if and only if $j_1 < \cdots < j_m$, and it is an SRT of shape $\delta / (1^n)$ if and only if $j_1 \geq \cdots \geq j_m$.*

Suppose now that T is an SRT of shape $\delta / (n)$. Then the invertibility of the jeu de taquin slides for SRTs along with the corollary above implies that T has been obtained from an SRT T' by doing a sequence of backward jeu de taquin slides as follows

$$T = \text{jdt}_{i_n} \cdots \text{jdt}_{i_1}(T'),$$

where $i_n > \cdots > i_1$. Similarly, if T is an SRT of shape $\delta / (1^n)$, then it has been obtained from an SRT T' by doing a sequence of backward jeu de taquin slides as follows

$$T = \text{jdt}_{i_n} \cdots \text{jdt}_{i_1}(T'),$$

where $i_n \leq \cdots \leq i_1$. Note now that the maps $\rho_{(n)} : \text{SSRT}(/(n)) \rightarrow \text{SSRCT}(/(n))$, and $\rho_{(1^n)} : \text{SSRT}(/(1^n)) \rightarrow \text{SSRCT}(/(1^n))$ are bijections. Using this and Theorem 6.21, it is clear that we can apply the above arguments to SRCTs. This insight is all we need to prove our right Pieri rule for the noncommutative Schur functions stated in Theorem 5.5.

9. A RIGHT PIERI RULE FOR NONCOMMUTATIVE SCHUR FUNCTIONS

To recover a Pieri rule from Theorem 3.6 that allows us to compute the products $\mathbf{s}_\alpha \cdot \mathbf{s}_{(n)}$ and $\mathbf{s}_\alpha \cdot \mathbf{s}_{(1^n)}$ we need to find all SRCTs of shape $\gamma // (n)$ (respectively $\gamma // (1^n)$) that rectify to the canonical tableau of shape α . In light of Corollary 8.11 we have the following lemma.

Lemma 9.1. *Let $\tau = \mu_{i_n} \cdots \mu_{i_1}(\tau_\alpha)$, where $i_n > \cdots > i_1$. Then τ rectifies to τ_α .*

Proof. Let $T_\alpha = \rho(\tau_\alpha)$ and $T = \rho_{(n)}(\tau)$. Then, by Theorem 6.21, we have that $T = \text{jdt}_{i_n} \cdots \text{jdt}_{i_1}(T_\alpha)$, which in turn implies that $\text{rect}(T) = \text{rect}(T_\alpha)$ (as discussed before Algorithm 2.10). Therefore, we get that τ rectifies to τ_α . \square

Remark 9.2. *The arguments after Corollary 8.12 imply that if there is an SRCT of shape $\gamma \parallel (n)$ that rectifies to τ_α , then it has to be obtained from τ_α by doing a sequence of backward jeu de taquin slides from columns i_1, i_2, \dots, i_n in that order, where $i_n > \dots > i_1$.*

Similarly, if there is an SRCT of shape $\gamma \parallel (1^n)$ that rectifies to τ_α , then it has to be obtained from τ_α by doing a sequence of backward jeu de taquin slides from columns i_1, i_2, \dots, i_n in that order, where $i_n \leq \dots \leq i_1$.

We can now prove the right Pieri rule, stated in Theorem 5.5.

Theorem 9.3. *(Right Pieri rule) Let α be a composition and n a positive integer. Then*

$$\begin{aligned} \mathbf{s}_\alpha \cdot \mathbf{s}_{(n)} &= \sum_{\substack{\beta \models |\alpha| + n, u_{i_n} \cdots u_{i_1}(\alpha) = \beta \\ i_n > \dots > i_1}} \mathbf{s}_\beta, \\ \mathbf{s}_\alpha \cdot \mathbf{s}_{(1^n)} &= \sum_{\substack{\beta \models |\alpha| + n, u_{i_n} \cdots u_{i_1}(\alpha) = \beta \\ i_n \leq \dots \leq i_1}} \mathbf{s}_\beta. \end{aligned}$$

Proof. We will only give the proof for $\mathbf{s}_\alpha \cdot \mathbf{s}_{(n)}$ as the proof of $\mathbf{s}_\alpha \cdot \mathbf{s}_{(1^n)}$ is similar. Notice first that if we have two distinct sequences $i_n > \dots > i_1$ and $j_n > \dots > j_1$ such that

$$\begin{aligned} \tau_1 &= \mu_{i_n} \cdots \mu_{i_1}(\tau_\alpha), \\ \tau_2 &= \mu_{j_n} \cdots \mu_{j_1}(\tau_\alpha), \end{aligned}$$

then τ_1 and τ_2 are distinct SRCTs. This follows from the reversibility of backward jeu de taquin slides for SRTs, once we consider the images of τ_1 and τ_2 under the generalized ρ map of Subsection 3.2. By Corollary 8.11, we get that

$$\begin{aligned} \text{sh}(\tau_1) &= \gamma_1 \parallel (n) \text{ where } \gamma_1 = u_{i_n} \cdots u_{i_1}(\alpha), \\ \text{sh}(\tau_2) &= \gamma_2 \parallel (n) \text{ where } \gamma_2 = u_{j_n} \cdots u_{j_1}(\alpha). \end{aligned}$$

We claim that $\gamma_1 \neq \gamma_2$. But in view of Remark 4.3 and given that the sequences $i_n > \dots > i_1$ and $j_n > \dots > j_1$ are distinct, this immediately follows. Using Remark 9.2, we get that all summands \mathbf{s}_γ in $\mathbf{s}_\alpha \cdot \mathbf{s}_{(n)}$ are obtained by picking a sequence $i_n > \dots > i_1$ such that $\mu_{i_n} \cdots \mu_{i_1}(\tau_\alpha)$ is an SRCT, and then letting γ be the outer shape of this SRCT. This gives us a multiplicity free expansion in the noncommutative Schur basis for the product $\mathbf{s}_\alpha \cdot \mathbf{s}_{(n)}$ as stated in the claim. \square

We conclude by proving a generalization of Theorem 8.5. Instead of counting maximal chains starting from \emptyset and ending at some composition β , we consider maximal chains starting from a composition α and ending at a composition β where $\alpha <_r \beta$.

Theorem 9.4. *Let $f_{\alpha,\beta}$ denote the number of maximal chains from α to β in \mathcal{R}_c . Then*

$$f_{\alpha,\beta} = \sum_{\gamma \models |\beta| - |\alpha|} C_{\alpha\gamma}^{\beta} f_{\emptyset,\gamma}.$$

Proof. Let $|\alpha| = m$ and $|\beta| = n + m$. If $\alpha <_r \beta$, then there is at least one sequence of positive integers i_n, \dots, i_1 such that $u_{i_n} \cdots u_{i_1}(\alpha) = \beta$. Let $w = i_n \cdots i_1$ consider the SRCT $\tau = \mu_{i_n} \cdots \mu_{i_1}(\tau_\alpha)$. Then $\text{sh}(\tau) = \beta // \gamma$ where $\gamma = \text{sh}(\rho^{-1}(Q(\text{std}(w))))$, by Theorem 8.10. Note also that τ rectifies to τ_α , by an argument similar to the proof of Lemma 9.1.

As stated in Theorem 3.6, we have that

$$C_{\alpha\gamma}^{\beta} = \text{number of SRCTs of shape } \beta // \gamma \text{ that rectify to } \tau_\alpha.$$

Now consider any skew SRCT τ of shape $\beta // \gamma$ that rectifies to τ_α . This can be completed to an SRCT τ_{aug} of straight shape β where the inner shape corresponding to γ has been filled according to an SSRCT of shape γ with distinct entries from the set $B = \{n + m, n + m - 1, \dots, m + 1\}$. Now, notice that every choice of the filling of shape γ gives a unique maximal chain from α to β . To see this, perform a forward jeu de taquin slide starting from the lower left corner of $\rho(\tau_{\text{aug}})$. As long as the resulting tableau has elements from B , repeat the previous step. Notice that the procedure above stops when we reach $\rho(\tau_\alpha)$. Reversing these slides starting from $\rho(\tau_\alpha)$, and using the generalized ρ map, defined in Subsection 3.2, at every step gives us the unique maximal chain from α to β .

Using the fact that the number of SRCTs of shape γ is $f_{\emptyset,\gamma}$, as established in Theorem 8.5, the claim follows. \square

Remark 9.5. *It is worth emphasizing that $f_{\alpha,\beta}$ is not the number of SRCT of shape $\beta // \alpha$ in general, although it is true in the case where α is \emptyset by Theorem 8.5. The above theorem can be considered as a noncommutative generalization of extracting coefficients in the classical identity $s_{\lambda/\mu} = \sum_{\nu \vdash |\lambda| - |\mu|} c_{\mu\nu}^{\lambda} s_{\nu}$.*

APPENDIX

We will outline the relation between our variant of the Robinson-Schensted algorithm and the classical Robinson-Schensted algorithm. The latter assigns to a permutation $\sigma \in \mathfrak{S}_n$ a pair of standard Young tableaux of the same shape and size equalling n . We begin by defining standard Young tableaux.

Given a partition λ , a *standard Young tableau (SYT)* T of *shape* λ is a filling of the boxes of λ with distinct positive integers from 1 to $|\lambda|$, satisfying the condition that the entries in T are strictly increasing along each row read from left to right and strictly increasing along each column read from bottom to top.

The classical Robinson-Schensted algorithm and evacuation action for SYTs (which we will denote by evac) are discussed in detail in [17, 20] and we will not present them here. To link the classical algorithms for SYTs with the variant for SRTs outlined in Subsection

2.5, we will need the notion of the complement of a permutation, and the complement of an SYT/SRT.

Given a permutation $\sigma = \sigma(1) \cdots \sigma(n)$ in single line notation, the *complement* of σ , denoted by σ^c is obtained by replacing $\sigma(i)$ by $n + 1 - \sigma(i)$ for $1 \leq i \leq n$. Similarly, given an SYT (SRT) T of shape $\lambda \vdash n$, the *complement* of T is the SRT (SYT), denoted by T^c , obtained by replacing an entry k in T by $n + 1 - k$. Finally, define T^t to be the transpose of any tableau T .

Observe that the complement and transpose operators on SYTs/SRTs commute. Furthermore, both the evac operator on SYTs and the e operator on SRTs commute with the transpose operator. Finally, note that

$$\begin{aligned} e(T) &= (\text{evac}(T^c))^c \text{ if } T \text{ is an SRT,} \\ \text{evac}(T) &= (e(T^c))^c \text{ if } T \text{ is an SYT.} \end{aligned}$$

Since evac is an involution by [20, Proposition A.1.2.9], we can see that e is also an involution.

Now we are ready to make explicit the relation between the classical algorithm for SYTs and the one for SRTs. Let $\pi \in \mathfrak{S}_n$. We will denote the insertion and recording tableaux obtained by using the classical Robinson-Schensted correspondence by $P(\pi)$ and $Q(\pi)$ respectively, and denote those obtained by using our variant by $P_v(\pi)$ and $Q_v(\pi)$ respectively.

Observe that

$$(P_v(\pi), Q_v(\pi)) = ((P(\pi^c))^c, (Q(\pi^c))^c).$$

Further using [20, Theorem A.1.2.10 and Corollary A.1.2.11] and the fact that evac is an involution, we get that

$$(P(\pi^c), Q(\pi^c)) = (\text{evac}(P(\pi)^t), Q(\pi)^t).$$

The two equalities above together imply that

$$(P_v(\pi), Q_v(\pi)) = ((\text{evac}(P(\pi)^t))^c, (Q(\pi)^t)^c).$$

Using [20, Corollary A.1.2.11] and the fact that $P(\pi^{-1}) = Q(\pi)$ we get that

$$\begin{aligned} P_v(\pi^{-1}) &= (\text{evac}(P(\pi^{-1})^t))^c \\ &= (\text{evac}(Q(\pi)^t))^c. \end{aligned}$$

Since $(Q_v(\pi))^c = Q(\pi)^t$, we get that

$$\begin{aligned} P_v(\pi^{-1}) &= (\text{evac}(Q_v(\pi)^c))^c \\ &= e(Q_v(\pi)). \end{aligned}$$

Again, using [20, Corollary A.1.2.11] and the fact that $Q(\pi^{-1}) = P(\pi)$ we get that

$$\begin{aligned} Q_v(\pi^{-1}) &= (Q(\pi^{-1})^t)^c \\ &= (P(\pi)^t)^c. \end{aligned}$$

Now, since $P(\pi)^t = \text{evac}(P_v(\pi)^c)$, we get that

$$\begin{aligned} Q_v(\pi^{-1}) &= (\text{evac}(P_v(\pi)^c))^c \\ &= e(P_v(\pi)). \end{aligned}$$

This establishes Lemma 2.12.

REFERENCES

- [1] M. BECHARD, *Rectification of Composition Tableaux*, arxiv preprint, <http://arxiv.org/abs/1201.4502>.
- [2] C. BESSENRODT, K. LUOTO, AND S. VAN WILLIGENBURG, *Skew quasisymmetric Schur functions and noncommutative Schur functions*, Adv. Math., 226 (2011), pp. 4492–4532.
- [3] W. FULTON, *Young Tableaux*, L.M.S Student Texts, Vol. 35, Cambridge University Press, Cambridge, 1997.
- [4] I. GELFAND, D. KROB, A. LASCOUX, B. LECLERC, V. RETAKH, AND J.-Y. THIBON, *Noncommutative symmetric functions*, Adv. Math., 112 (1995), pp. 218–348.
- [5] J. HAGLUND, M. HAIMAN, AND N. LOEHR, *A combinatorial formula for Macdonald polynomials*, J. Amer. Math. Soc., 18 (2005), pp. 735–761.
- [6] J. HAGLUND, K. LUOTO, S. MASON, AND S. VAN WILLIGENBURG, *Quasisymmetric Schur functions*, J. Combin. Theory Ser. A, 118 (2011), pp. 463–490.
- [7] M. HAIMAN, *On mixed insertion, symmetry, and shifted Young tableaux*, Journal of Combinatorial Theory. Series A, 50 (1989), pp. 196–225.
- [8] A. LASCOUX AND M.-P. SCHÜTZENBERGER *Le monoïde plaxique, Noncommutative structures in algebra and geometric combinatorics (Naples, 1978)*, Quad. “Ricerca Sci.” Vol. 109, CNR, Rome, 1981, pp. 129–156.
- [9] M. VAN LEEUWEN, *An analogue of jeu de taquin for Littelmann’s crystal paths*, Sémin. Lothar. Combin., 41 (1998).
- [10] K. LUOTO, S. MYKYTIUK, AND S. VAN WILLIGENBURG, *An introduction to quasisymmetric Schur functions. Hopf algebras, quasisymmetric functions, and Young composition tableaux.*, Springer Briefs in Mathematics, Springer, New York, 2013.
- [11] I. MACDONALD, *Symmetric functions and Hall polynomials. 2nd ed.*, Oxford University Press, 1998.
- [12] S. MASON, *A decomposition of Schur functions and an analogue of the Robinson-Schensted-Knuth algorithm*, Sémin. Lothar. Combin., 57 (2008).
- [13] S. PON AND Q. WANG, *Promotion and evacuation on standard Young tableaux of rectangle and staircase shape*, Electron. J. Combin., 18 (2011).
- [14] R. PROCTOR, *d-Complete posets Generalize Young Diagrams for the Jeu de Taquin Property*, arxiv preprint, <http://arxiv.org/abs/0905.3716>.
- [15] L. RIEGLER AND C. NEUMANN, *Playing jeu de taquin on d-complete posets*, arxiv preprint, <http://arxiv.org/abs/1401.3619>.
- [16] B. SAGAN, *Shifted tableaux, Schur Q-functions, and a conjecture of R. Stanley*, J. Combin. Theory Ser. A, 45 (1987), pp. 62–103.
- [17] B. SAGAN, *The symmetric group. Representations, combinatorial algorithms, and symmetric functions. 2nd ed.*, Springer, 2001.
- [18] M.-P. SCHÜTZENBERGER, *La correspondance de Robinson, Combinatoire et représentation du groupe symétrique (Actes Table Ronde CNRS, Univ. Louis-Pasteur Strasbourg, Strasbourg, 1976)*, Lecture Notes in Math. Vol. 579, Springer, Berlin, 1977, pp. 59–113.
- [19] P. ŚNIADY, *Robinson-Schensted-Knuth algorithm, jeu de taquin, and Kerov-Vershik measures on infinite tableaux*, SIAM J. Discrete Math., 28 (2014), pp. 598–630.

- [20] R. STANLEY, *Enumerative Combinatorics*, vol. 2, Cambridge University Press, 1999.
- [21] R. STANLEY, *Promotion and evacuation*, Electron. J. Combin., 16 (2009).
- [22] H. THOMAS AND A. YONG, *A jeu de taquin theory for increasing tableaux, with applications to K-theoretic Schubert calculus*, Algebra & Number Theory, 3 (2009), pp. 121–148.
- [23] H. THOMAS AND A. YONG, *Equivariant Schubert calculus and jeu de taquin*, arxiv preprint, <http://arxiv.org/abs/1207.3209>.

VASU V. TEWARI, DEPARTMENT OF MATHEMATICS, UNIVERSITY OF BRITISH COLUMBIA, VANCOUVER,
BC V6T 1Z2, CANADA

E-mail address: vasu@math.ubc.ca